The Dissertation Committee for Lars Gustaf Omberg
certifies that this is the approved version of the following dissertation:

Tensor Generalizations of the Singular Value Decomposition for Integrative Analysis of Large-Scale Molecular Biological Data

Committee:

Orly Alter, Supervisor

Greg O. Sitz, Supervisor

Ernst-Ludwig Florin

Vishy Iyer

Edward M. Marcotte

Michael P. Marder

Tensor Generalizations of the Singular Value Decomposition for Integrative Analysis of Large-Scale Molecular Biological Data

by

Lars Gustaf Omberg, MsE

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December 2007

To my family: My siblings, Jonas and Karin, for being positive role models; my parents, Gunnar and Camilla, for being supportive even when I was being foolish; and my aunts, Ankan and Babben, for providing support when my parents were too far away.

Tensor Generalizations of the Singular Value Decomposition for Integrative Analysis of Large-Scale Molecular Biological Data

Publication No. _____

Lars Gustaf Omberg, Ph.D.
The University of Texas at Austin, 2007

Supervisors: Orly Alter and Greg O. Sitz

The structure of large-scale molecular biological data is often of an order higher than that of a matrix, especially when integrating data from different studies. Flattened into a matrix format, much of the information in the data is lost. I describe the use of higher-order generalizations of singular value decomposition (SVD) - both the higher-order singular value decomposition (HOSVD) and Parallel Factorization (PARAFAC) - in transforming tensors into simplified spaces. I apply these transformations to a series of DNA microarray datasets from different studies tabulated

in a tensor of genes × time × conditions, specifically an integration of genome-scale mRNA expression data from three yeast-cell cycle time courses. One of the time courses was under exposure to the oxidative stress agent hydrogen peroxide (HP); another was exposed to menadione (MD) and the third was unstressed[45].

The HOSVD transforms the tensor to a "core tensor" of "eigenarrays" × "time-eigengenes" × "condition-eigengenes," where the eigenarrays, time-eigengenes and condition-eigengenes are unique orthonormal superpositions of the genes, times and conditions, respectively. This HOSVD, also known as N-mode SVD, formulates the tensor as a linear superposition of all possible outer products of an eigenarray, a time-eigengene and a condition-eigengene, i.e., rank-1 "subtensors," the superposition coefficients of which are tabulated in the core tensor. Each coefficient indicates the significance of the corresponding subtensor in terms of the overall information it captures in the data. PARAFAC reformulates the same data tensor into a sum of rank-1 tensor of F elements that best approximate the data tensor in a least square sense.

I show that significant rank-1 subtensors can be associated with independent biological processes, which are manifested in the data tensor. Subtensors of the HOSVD capture the subprocesses: stress response, pheromone response and developmental stage. The data suggests that the conserved genes YKU70, MRE11, AIF1 and ZWF1, as well as the genes involved in the processes of retrotransposition, apoptosis and the oxidative pentose phosphate cycle may play significant, yet previously unrecognized, roles in the differential effects of HP and MD on cell cycle progression. Subtensors of PARAFAC capture the same biological processes as the 2 most significant HOSVD subtensors. A genome-wide correlation between DNA replication and initiation of RNA transcription, which is equivalent to a recently discovered correlation and might be due to a previously unknown mechanism of regulation, is independently uncovered.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The advent of high throughput techniques in molecular biology has sprouted a revolution in data availability. The amount of data is growing exponentially but the techniques to handle them are not. Much remains unanalyzed or only partially analyzed. Frameworks to bridge this gap are necessary. Frameworks that will not only allow for prediction and hypothesis generation, from the data, but also to drive discovery and model biology for possible ultimate control would be preferable [5]. Judging from the rapid pace of new technologies being adapted, the data currently available is probably quite different from the data that will be available in the future, so generality would also be preferable.

Every new high-throughput technique has caused a separate revolution that is slowly elucidating the inner workings of cells and organisms. One of the first such revolutions stemmed from the sequencing of whole genomes which slowly allowed coding regions to be identified. Later microarrays were developed, giving birth to functional genomics. Now we are moving into the post-genome era, beyond transcriptomics, gene lists, and functional genomics to proteomics. Advances in mass spectroscopy now allow quantitative measurement of protein quantities within cells, a develop-

ment that is already seeing clinical application in biomarker fingerprinting for cancer detection [46]. New generation sequencing technologies such as massively parallel pyrosequencing, which are capable of producing tens of millions of sequence reads in a matter of hours, are being applied in creative ways to answer genome-wide questions [41].

All these data and very likely the data available in the future will be presentable in similar ways, as arrays of numbers. Some of the arrays are best represented as tensors. In this thesis I will focus on microarray data because it is a relative mature technology with more data available in public databases than any of the other data mentioned.

DNA microarrays make it possible to record genome-scale signals, such as, mRNA expression levels [53, 56, 51, 26] and proteins' DNA-binding occupancy levels [28, 67, 54], that guide the progression of cellular processes. Future discovery and control in biology and medicine will come from the mathematical modeling of these data, where the mathematical variables and operations represent biological reality. The variables, patterns uncovered in the data, might correlate with activities of cellular elements, such as regulators or transcription factors, that drive the measured signals. The operations, such as data classification and reconstruction in subspaces of selected patterns, might simulate experimental observation of the correlations and possibly also causal coordination of these activities [5]. Comparative analyzes of these data among two or more organisms might give insights into the universality and specialization of evolutionary, biochemical, and genetic pathways [6]. Integrative analyses of different types of signals from the same organism might reveal cellular mechanisms of regulation [7].

The structure of DNA microarray data is usually seen as a matrix, with genes on one axis and experiments on another. When multiple experimental conditions are

varied the natural order is higher than that of a matrix. Each of the experimental conditions is best represented on its own axis. This is also true for integrative analysis of different studies where each of the multiple biological and experimental settings under which the data are measured represents a degree of freedom in a tensor [3]. Unfolded into a matrix, these degrees of freedom are lost and much of the information in the data tensor might also be lost. Furthermore the interpretation of the flattened data can be difficult as the effects of the different experimentally applied parameters will be intermixed (Fig. 1.1) [11].

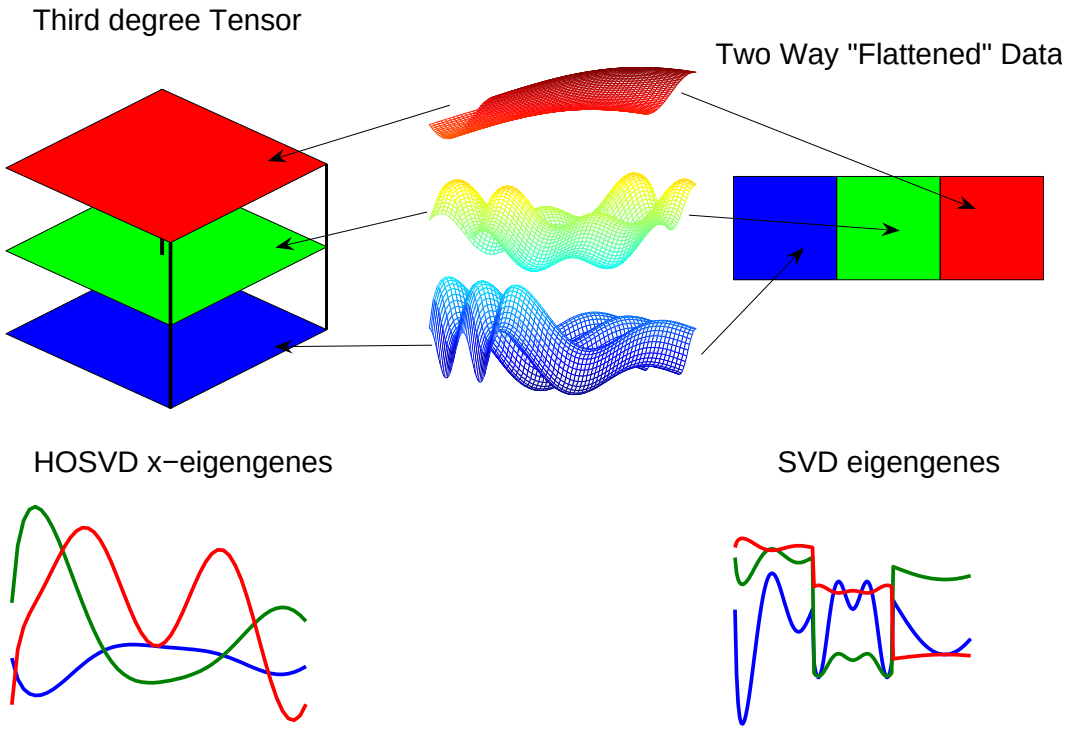


Figure 1.1: Comparison of two-way Singular Value Decomposition to Higher Order Singular Value Decomposition.

I describe the use of higher order generalizations of the matrix singular value decomposition (SVD) to data tensors.

The higher order singular value decomposition (HOSVD) also known as n-mode

singular value decomposition [18, 35, 68] transforms a data tensor of genes × "x-settings," that is, different settings of the experimental variable × "y-settings," which tabulates DNA microarray data from different studies, to a "core tensor" of "eigenarrays" "x-eigengenes" "y-eigengenes." The eigenarrays and x- and y-eigengenes are unique orthonormal superpositions of the arrays and the genes across the x- and y-settings, respectively. I Reformulate this multilinear HOSVD [27, 1, 4] such that it decomposes the data tensor into a linear superposition of all outer products of an eigenarray, an x- and a y-eigengene, that is, rank-1 "subtensors" [18]. The superposition coefficients, of this linear superposition, are the "higher-order singular values" tabulated in the core tensor which define the significance of each subtensor in terms of the fraction of the overall information in the data tensor that the subtensor captures.

The parallel factorization (PARAFAC)[9, 11] is another generalization that reformulates the same data into a fixed sized diagonal core tensor of "arrayfactors"× "x-genefactors"×"y-genefactors" where the reformulation approximates the original data in a least square sense. The number of factors and, hence. the size of the core tensor is determined a priori. The diagonal nature of the core tensor means that each arrayfactor is only associated with one x-genefactor and one y-genefactor, meaning that the decomposition can be reformulated as a sum of rank-1 subtensors. This simplifies the interpretation of this technique, but, as a model for the kind of data I have examined it seems too restrictive.

I illustrate these higher-order generalization of SVD with an integration of genome-scale mRNA-expression data from three yeast-cell cycle-time courses, two of which were exposed to either hydrogen peroxide (HP) or menadione (MD) [53, 56], two oxidative stress agents. I found that significant subtensors represent independent biological programs or experimental phenomena common to all three studies or exclusive to either one or two of the studies [58]. This includes the subtle differential

effects of HP and MD on cell cycle progression. I also found that this subtensor interpretation is robust to variations in the data selection cutoffs. The picture that emerges from this data-driven analysis suggests that the conserved genes YKU70, MRE11, AIF1, and ZWF1 and the processes in which they are involved in - retrotransposition, apoptosis and the oxidative pentose phosphate pathway - may play significant, yet previously unrecognized, roles in the differential effects of HP and MD on cell cycle progression [53, 25, 38, 22, 52, 43, 16, 66, 37, 36]. A genome-scale correlation between DNA replication initiation and RNA transcription, which is equivalent to a recently discovered correlation [7], is consistent with the current understanding of replication initiation [42, 20, 15, 8] and recent experimental results [21, 55, 50, 12, 14], and might be due to a previously unknown mechanism of regulation, is independently uncovered.

Chapter 2

The Ten Minute Introduction to Biology

The biology necessary to understand this treatise is very limited. None of the material I present in the next few pages is above what is usually covered in high school biology and can be skipped without loss of continuity. It all pertains to the idea of information flow in the genetic material of the cell.

There are three major biopolymers in cells - DNA, RNA and proteins. Proteins are involved in every vital process of the cell such as catalysis of chemical processes, structural support, transport, replication, immune response etc. A protein is made of amino acids forming long chains the order of which are defined by a gene. The gene is encoded in the DNA. DNA is the long-term information-storage component of the cell; in simplified terms it is often called the "blueprints of the cell." RNA plays several important roles in the process that translates the DNA into proteins. These biopolymers are all linear, meaning that each monomer is connected to a maximum of two other monomers. The order and type of monomer determines the

properties of the entire polymer.

2.1 DNA is the carrier of genetic information

DNA serves two purposes for information transfer: replication and transcription. Replication is the process by which the DNA is duplicated and passes genetic information from parent to progeny. Transcription is the process by which genes - the regions of DNA containing genetic information that can affect the phenotype of the organism - are transcribed, by RNA polymerase into the related nucleic acid, RNA.

Chemically DNA is made up of a series of simple units known as nucleotides. which have a backbone of sugars and phosphate atoms. Each sugar is attached to one of four different bases: adenine (A), cytosine (C), guanine (G) and thymine (T). The order of these bases encodes the instructions for forming all other cellular components of the organism. Within the cell, the DNA polymer forms a helix together with a complementary strand in which the nucleotides of the strands bond to each other by a process called complementary base pairing or hybridization, such that C always binds to G, and A always binds to T. This means that each strand duplicates the information of the other strand. This duplication is central for the replication and transcription of DNA.

2.2 RNA is the messenger for production of proteins

RNA is similar to DNA with a few structural differences. The nucleotide thymine is most often replaced by uracil (U) but otherwise the information is stored in the same manner as in DNA. Most of the RNA is used to synthesize proteins by either acting as templates for the translation of genes into proteins or by transferring amino acids to the ribosome.

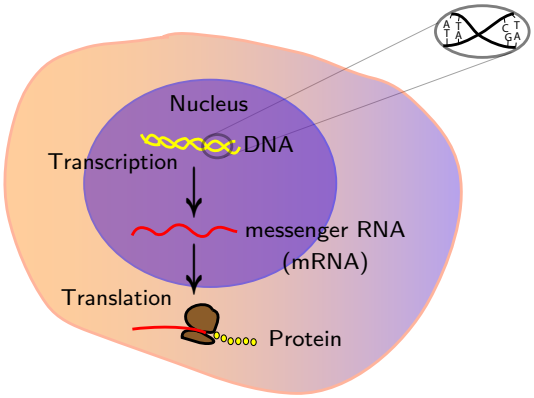


Figure 2.1: Central Dogma

Ribosomes read the messenger RNA (mRNA) and manufacture proteins by a process called translation. Information in the mRNA is encoded by codons, consisting of a triplet of nucleotides where each codons codes for a specific amino acid, the signal for start translation or stop translation. This code of translation determines the chain of amino-acids that are created from each mRNA. This polypeptide, sometime after post-translational modifications, becomes protein.

The simple rules governing the transcription of DNA to mRNA and transcription of mRNA into proteins is sometimes called the central dogma of molecular biology, a term coined by Francis Crick in 1958 [17]. While not entirely true in its original formulation - which stated that information never flowed backwards from proteins to RNA to DNA something that we today actually believe does happen for regulatory purposes - is still good for illustrative purposes. Regulation is essential. When a cell requires a certain protein, the gene encoding that protein will be activated and mRNA will be transcribed and then translated. By measuring the amount of

protein and/or mRNA of a specific gene one can try to determine which genes are important for specific processes.

2.3 Functional Genomics

In the early 1970s the sequences of individual genes were beginning to be identified. With the advent of rapid sequencing techniques recent years have seen complete genome sequences identified and the birth of the field of genomics. As of September 2007 the National Center for Biotechnology Information (NCBI) lists the complete sequence of about 1879 viruses, 577 bacterial species and roughly 23 eukaryote organisms including humans [65]. Though, this information alone is not enough to understand the functioning of an organism, it allowed for the birth of the field of functional genomics. Functional genomics is a field of study that is mainly concerned with gene expression under different conditions, in order to describe gene and protein functions and interactions and the dynamics of these interactions and expressions.

One of the tools that has become heavily relied upon in functional genomics is DNA-micorarrays, which is also the focus of this dissertation. It is worth noting that almost none of the techniques mentioned herein are specific to microarrays and are easily transferable to other large-scale data such as proteomics.

2.4 Microarrays

A DNA microarray is both a technique and an experimental device consisting of a collection of tiny DNA spots in which each spot contains multiple copies of identical sequences of single-stranded DNA. Often these spots represent single genes, but they can be any sequence. By using micro-pipettes, photo-lithography or ink-jet printing techniques, thousands of spots can fit onto a single cm^2 sized chip and,

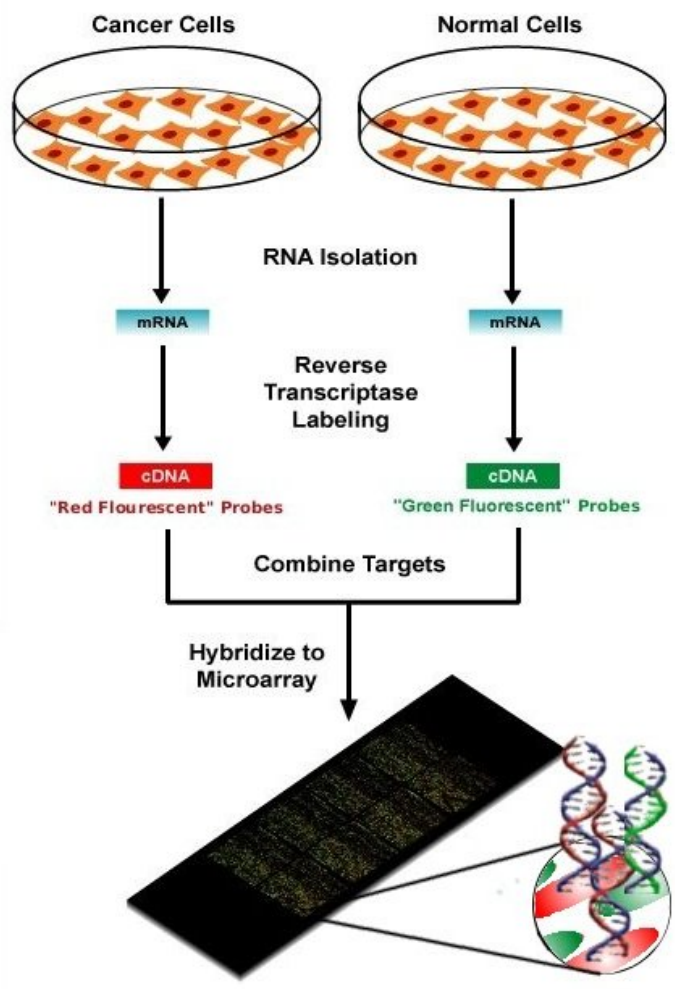


Figure 2.2: Schmeatic drawing of typical micorarray experiment using a spoted array

therefore, can represent the entire genome of an organism. Measurements are made possible by the specificity in hybridization between complementary strands of DNA. Actually complimentary base pairing coupled with fluorescently labeled DNA for reading using laser excitation are the fundamentals of the technology.

Microarrays allow quantitative measurements of amounts of DNA, but they can also measure other things by using DNA as an intermediate step in experiments such as comparative genomic hybridization (e.g., [47, 48]), single nucleotide polymorphism (SNP) detection (e.g., [60, 57]) and chromatin immunoprecipitation (chIP) studies (chIP-chip) (e.g. [31, 28]). The most common type of experiment to use microarrays is expression profiling, in which the amount of mRNA is measured under different physiological conditions.

As cells live they require different proteins to maintain homeostasis; for example when diseased different genes will be expressed than when healthy. By measuring the amount of mRNA present of each type of gene under different conditions it is possible to determine which genes are important for each condition. A very simple experiment that does just that is diagrammed in figure 2.2 for a spotted microarray or two-channel microarray. The microarray has been printed with each spot representing a unique gene. Two samples of cells that are to be compared are harvested and the mRNA isolated. The two samples of mRNA are separately reverse-transcribed to differently fluorescently labeled cDNA (e.g., Cyanine 5 for red and Cyanine 3 for green). The samples are then mixed and competetively hybridized to the microarray. After washing, the microarray is read by stimulating the fluorophor with a laser and reading the emission of each spot. The relative intensity of each fluorophor gives a measurement of relative up/down regulation of each gene in the two different samples.

Chapter 3

Useful Techniques for Analyzing Genome Scale Data

When analyzing large-scale molecular biological data, many analysis techniques and steps are carried out multiple times both before analysis and after. This chapter is dedicated to those techniques and the tools I have made to deal with them. Specifically, when analyzing functional genomic data the process can be divided into four steps: 1, data acquisition; 2, data preprocessing; 3, data analysis; and 4, interpretation. The first, second and fourth steps will be discussed in this chapter and are generic for most datasets, while step three will be expounded in chapter 4 when I will introduce tensor analysis. All steps will be combined in chapter 5 where the result of analysis will be shown.

Data acquisition is the retrieval of data from databases and the assessment of data quality. Data acquisition is also the step in which data is filtered. Filtering is a factor that results in missing data, which means preprocessing will be necessary, which consists of further filtering or estimating missing values. Functional association is a

Stanford Microarray Database (SMD) [a]
NCBI Gene Expression Omnibus [b]
EBI ArrayExpress [c]
ExpressDB [d] [e]
yeast Microarray Global Viewer(yMGV) [f] [g]
Center for Information Biology gene EXpression (CIBEX) [h]

[a] http://genome-www5.stanford.edu/
[b] http://www.ncbi.nlm.nih.gov/geo/
[c] http://www.ebi.ac.uk/microarray-as/aer/
[d] No new data anded in at least a year
[e] http://arep.med.harvard.edu/ExpressDB/
[f] No new data anded in at least a year
[g] http://www.transcriptome.ens.fr/ymgv/
[h] http://cibex.nig.ac.jp/index.jsp

Table 3.1: List of Microarray Databases that store publicly searchable data

very important tool for interpretations before and after data analysis.

3.1 Data retrieval and filtering

Gene expression data consisting of the abundance of RNA or DNA of multiple specific sequences is usually stored in databases. Due to the common requirement by journals for authors to publish the raw data along with articles, several public databases have been created. Some of the more common of these are listed in table 3.1. The data is retrievable in many forms depending on the microarray platform and source database. I will focus on the typical retrieval of two color microarrays from databases such as the Stanford Microarray Database (SMD)[19] or the freeware version the Longhorn Array Database (LAD)[33].

In these databases data filtering can be carried out with multiple statistical methods and stringency on the data quality etc. A database can however take up to an hour to return a large dataset. In order to explore the stability of results it is desirable to try multiple conditions. To avoid having to visit a database and input multiple

filtering options and download the data multiple times I implemented a local data structure and command-line tool to do most of the important filtering locally.

The premise behind this tool is to apply some of this filtering locally where it can easily be tried multiple times. The manual for this tool is available in appendix A.2.1 along with a sample usage that will extract all the data used for analysis in chapter 5. Some of the main functions will be highlighted below.

3.2 Filtering

One microarray with I_1 probes is represented as a vector of length I_1. A set of experiments with I_2 microarrays can be combined into a matrix, D, where each column, $T_{:n}$, represents one experimental condition and each row $T_{m:}$ represents the expression profile of a gene (or probe) across the I_2 conditions. During experiments some of these values in the matrix will be nonexistent.

Nonexistent values can be due to experimental artifacts such as scratches or dust on the microarray chip or chemical or biological discrepancies during the hybridization process. Experimenters can also classify spots as missing if during the scanning process the fraction of pixels within the spot that are brighter than the median of the background are below a threshold or the intensity of a spot is below a certain threshold. This will often lead to specific microarrays (conditions) or genes that have a disproportionally large number of missing values. A common step at this point is to remove these genes or experiments. That is, filter out those columns for which

$$\sum_{i=1}^{I_1}(T_{in} == \text{NaN}) > t$$

or rows for which

$$\sum_{j=1}^{I_2}(T_{mj} == \text{NaN}) > t$$

where t is the threshold number of missing values allowed and NaN is the standard symbol for "not a number" or in this case missing value.

With this type of filtering it is possible to eliminate all missing values by setting $t = 0$ but for other values of t it is often necessary to estimate those missing values.

3.3 Missing Data Estimation

Most analysis of data, be they derived from statistics, signal processing or biology require complete datasets without missing values. Several techniques have been developed to handle missing values. Most however do not stem from bioinformatics and very few methods have been developed specifically for this purpose [61]. As the popularity of microarrays has increased, the methods used for estimating missing values have become more complex. The easiest method of replacing missing values is to replace them with zeros, something that is especially defensible for log ratio data. Slightly more complicated is to replace missing value T_{mn} in the matrix by the row average

$$T_{mn} = \frac{\sum_{j\in\{\text{non missing}\}} T_{mj}}{|\{\text{non missing}\}|}$$

where {non missing} is the set of values in row m that are not missing. Though simple, this method has been shown to be best for replacing values when the experiments do not have any correlation structures[24].

Several more complicated methods have been applied to microarray data such as the popular K Nearest Neighbor Imputation [61], Local Least Square Imputation [34]

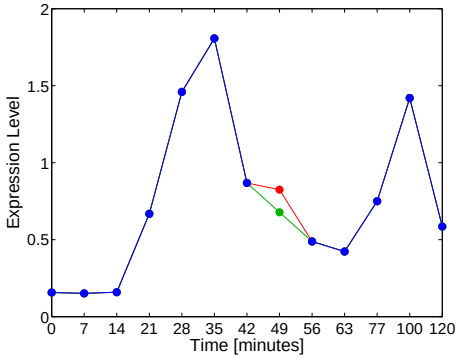


Figure 3.1: Comparison of two different imputation methods. The blue line represents the expression pattern of the hi stone gene Htb2 for time series experiment [56]. The measurement at 49 minutes is missing and has been replaced with the linear interpolation(green) and SVD imputed value(red).

and Bayesian Principal Component Analysis [44]. I will explain two specific methods: Interpolation and singular value decomposition Imputation, for an example of a missing value and the resulting imputation see figure 3.1. This is justified by two observations: different methods' ability to predict missing values is dependent on the data [24] and that for time series data with low noise levels SVDimpute works well and better than KNNimpute[61], which many people consider the standard method.

3.3.1 Linear Interpolation

If the multiple conditions represent a succession of measurements where a single variable is varied in a predictable pattern such as a time series or a series of concentration experiments, the correlation between values in the columns allows imputation using interpolation. The most basic form is linear interpolation, which is trivial if only single values are missing and none of the values are in the first or last column.

$$T_{mn} = T_{mn-1} + \frac{T_{mn+1} - T_{mn-1}}{2}.$$

If values are located at the edges, one can use an extrapolation, assuming that the derivative of the expression with regards to measurement is constant. If it is not constant one can assume that the measurement in column 0 is the same as the measurement made after the last column. The latter method, also known as toroidal geometry, is what I implemented in the Microarray package. Neither of these assumptions is optimal, and the best choice is dependent on the particular experiment. If more than one value is missing adjacent to each other the linear interpolation is extended by assuming linear behavior for all the missing values.

3.3.2 Singular Value Decomposition Imputation

Another approach to imputing missing values is to create a model of all the data and then to use the model to estimate the missing values using a least square approximation. One such model is the singular value decomposition (SVD). It has been shown that the SVD allows experiments to be rewritten as the outer product between *eigenarrays* and *eigengenes* where they represent cellular states and independent processes respectively[1]. The high entropy[1] of microarray data, where most of the information is captured by a few significant eigenvectors, also allows for dimensional reduction of the model by assuming the less significant eigenvectors insignificant. By using this property a reduced model is constructed and the missing values can be estimated from this model [2].

Specifically the model is constructed by calculating the SVD of the reduced data set where all the probes or rows of the matrix with missing values are removed. This

[1]By entropy I mean Shannon entropy as defined in equation 4.3

creates the reduced matrix $T' \in \mathbb{R}^{I_1' \times I_2}$ with the SVD decomposition,

$$T' = U\Sigma V^T.$$

This will give $\min(I_1', I_1)$ eigengenes of which L are significant. A linear superposition of these L eigengenes is then assumed to represent any probe. That is $T' \approx U'\Sigma' V'^T$ where $U' \in \mathbb{R}^{I_1' \times L}$, $V' \in \mathbb{R}^{L \times L}$ and $\Sigma \in \mathbb{R}^{L \times L}$

All missing values in the $I_1 - I_1'$ probes, $T_{i:}$ are computed by approximating the values of $T_{i:}$ that are not missing with a superposition of the L eigengenes. That is the matrix equation $T_{i:}'' = V''c$ is solved in a least square sense for the L superposition coefficients c where $T_{i:}''$ contain only those positions without missing values and V'' contains the corresponding positions out of V'. That is

$$c = V''^{\dagger} T_{i:}''$$

where † represents the Rose-Penrose pseudo-inverse. The missing values are then imputed by $T_{ij} = (c \cdot V')_j$.

3.4 Normalization

Typically the first transformation done to data after missing values are estimated is normalization. The goal of normalization is to get the data into a format that allows meaningful biological comparisons by compensating for experimental inequalities between the arrays. These inequalities can be caused by differences in: quantity of starting RNA, spot quality, fluorescent dyes, and protocol, to name a few. There are also normalization techniques that are applied within individual arrays to handle differences in printing between different tips [49], but this use will be not considered

in this text. The primary use of these three normalizations is to simplify integrative analysis.

3.4.1 Array Centering

In two-color arrays the two competing hybridizations can show a bias caused by either a difference in the dyes, difference during scanning or experimental bias in the extraction of the RNA for one of the colors. To correct for this each array can be re-centered by subtracting the mean:

$$T'_{:j} = T_{:j} - \sum_{i=1}^{M} \frac{T_{ij}}{M}$$

This normalization is valid for both ratio and log ratio data and could equally well be applied to single-color arrays.

3.4.2 Array Scaling

The total intensity of arrays or the dynamic range of the arrays can be different. This intensity difference can be biological in nature but more often than not it is an experimental artifact. To compensate, one can by scale the entire array by the norm. Using the l^2-norm each array becomes:

$$T'_{:j} = \frac{T_{:j}}{\sqrt{T_{:j} \cdot T_{:j}}}$$

This normalization, however, is not appropriate if there is an explained reason for the intensity of the arrays. In that case, a less stringent normalization is more appropriate.

3.4.3 Normalization by Frobenius Norm

The extension of array scaling to biases in multiple arrays is scaling by the Frobenius norm. When integrating multiple datasets the dynamic range of each dataset (not each array) should be compensated for. Completely in analogy with the vector norm used for array scaling the Frobenius norm is the square root of the sum of squares of all the elements in a matrix.

$$T' = \frac{T}{||T||_F}, \qquad ||T||_F = \sum_i \sum_j \sqrt{T_{ij}^2}$$

3.5 Calculation of Enrichment

When analyzing data many, techniques lead to grouping of probes by either internal similarity or similarity to a model. For example, when clustering probes with similar expression-patterns are grouped together into clusters. When calculating the SVD, the eigenarrays represent the strength of expression of all the probes in each of the eigengenes. This means that the most strongly expressed genes within an eigenarray give a group of genes most similar in expression to the model pattern of the associated eigengene. We can attempt to assign these groups to known biological programs by associating the probes with annotations. Using the hypergeometric distribution, I can obtain a measurement of the certainty of the associated biological program. That is, the groups are examined for enrichment of a certain annotation and the probability of that enrichment not occurring by chance is calculated from the hypergeometric distribution. To be able to explore these enrichments I built a framework and a command line tool (see appendices A.1, A.2.1 and A.3.2)

3.5.1 The Hypergeometric Distribution

In statistics the hypergeometric distribution is usually introduced as a way to describe the number of successes in sequence of m draws from a population of size N without replacement. If the total number of possible successes in the population is K, the probability of obtaining l successes is described by the distribution function

$$f(l; N, K, m) = \frac{\binom{K}{l}\binom{N-K}{m-l}}{\binom{N}{m}} \qquad \text{where} \qquad \binom{N}{m} = \frac{N!}{(N-m)!m!}.$$

This distribution is apparent from combinatorics alone. In the case of enrichment of genes, we consider the population to be the number of probes on the array or the number of genes in genome if the entire genome is present; the sample size is the number of genes in a cluster or group; a success is any gene labeled by the annotation that is being tested for enrichment. The measurement of the probability that the enrichment is not due to chance is the p-value given by[59]

$$P = \sum_{i=l+1}^{m} \frac{\binom{K}{i}\binom{N-K}{m-i}}{\binom{N}{m}}.$$

The calculation is very time consuming as the factorial is difficult to calculate.

Speeding up hypergeometric calculation

To simplify the calculation of the distribution, we can rewrite f(l;N,K,m) by inserting the binomial operator and simplifying terms obtaining

$$f(i, K, N, m) = \frac{K!(N-K)!(N-m)!m!}{(K-i)!i!(N-K-m+i)!(m-i)!N!} \tag{3.1}$$

and the P value is actually equivalent to

$$P = \sum_{i=l+1}^{m} e^{\ln(f(i,K,N,m))}. \tag{3.2}$$

Combining equations and with the basic properties of logs namely $\log(a * b) = \log(a) + \log(b)$ and $\log(a/b) = \log(a) - \log(b)$.

$$\ln(f(i,K,N,m)) = \ln(K!) + \ln((N-K)!) + ...$$

where if the argument to the log is large we can approximate it using Sterlings formula

$$\ln(n!) = n\ln(n) - n + \frac{1}{2}\ln(2\pi n) + \frac{1}{12n} - \frac{1}{360n^3} + ...$$

This process allows screenings, in practice, to be done four times as quickly without a sacrifice in precision.

3.5.2 Visualizing the enrichment

As the number of annotations grows, a new difficulty arises, namely visualization. Large numbers of annotations with similar p-values are difficult to distinguish and when annotations are similar clear trends become difficult to discern. Related to this is the difficulty in screening for the correct enrichment cutoff when looking at enrichment among probes that are characterized by their similarity to a model, which means the size of the sample is not obviously defined. The first problem can be solved by grouping annotations by similarity and color coding the annotations.

The latter problem is solved by quick overview graphics that allow us to explore the

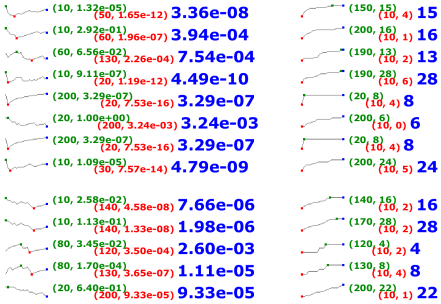


Figure 3.2: Example showing partial output of enrichment using sparklines. Each row represents one annotation. The two sparklines plot the p-value and number of successes in the sample respectively with the highest(green), lowest(red) and last(blue) values displayed.

data beyond the numerical limits alone. Edward Tufte created "data-intense, design-simple, word-sized graphics" known as sparklines[64]. Usually used as elements of a "small multiple" or a series of repetitive visual elements that together make a point, they allow us to explore the enrichment data. For each annotation the enrichment is calculated for multiple sample sizes and then the p-value and the number of successes are plotted over the sample size. If the probes are sorted by their similarity to a model, smaller samples will capture only the most strongly correlated probes while larger samples will capture fewer correlated probes. Figure 3.2 shows an example output of enrichment calculations including sparklines that was generated using the tool described in Appendix A.1.

3.5.3 Annotation Data

The annotations used for the enrichment calculations are culled from literature and databases. I have examined mostly yeast data but, to a lesser degree, also human data. In the case that the data comes from microarrays, I depend on classifications done by other experimenters. This usually means some kind of clustering was attempted and genes in specific clusters were identified to contain a certain property

or function. Other large-scale data stemming from Chromatin immunoprecipitation (ChIP) microarrays or ChIP-chip assays give locality of protein DNA interactions. This is important for such processes as transcription factor binding and DNA replication. This allowed genes to be upstream or near these sites to be classified as being influenced or affected by those proteins. Annotations obtained in this manner are described in detail in appendix B. Furthermore, annotations as defined by the Gene Ontology were examined.

Gene Ontology Annotations

The Gene Ontology (GO) is a semantic network providing a controlled vocabulary with which to define genes and gene products. To be strictly correct it provides three separate networks that each encompass separate concepts: cellular component or localization within the cell; molecular function; and role in biological processes. The vocabulary in the networks is linked by two types of relationships: is_a or part_of. Together the vocabularies form networks that are directed acyclic graphs. Directed acyclic graphs differ from hierarchies in that a child can have multiple parent terms. To illustrate, an engine is a machine but can also be classified by being a part of a car and because of the hierarchy it will be defined by all the parent terms of machine and car. The same is true for gene products classified by a GO term and also classified by all of the terms parents and their parents and so on. The second part of the Gene Ontology is the members of the GO consortium who assign GO terms to the gene products for several organisms.

It is the combination of both of these that allows us to observe the annotations of probes. Since each probe can have multiple GO terms assigned, the enrichment has to be combinatorially determined by traversing the entire Gene Ontology hierarchy. The tools used to do this are described in detail in appendix A.3.2.

Chapter 4

Generalizing Singular Value Decomposition to Tensors

Extending the singular value decomposition(SVD) to tensors or multidimensional arrays can be accomplished in multiple ways. Properties - such as rank, diagonality, orthogonality - that have clear definitions and simple relationships in matrix algebra have multiple definitions or complicated relationships in tensor algebra. There are two major generalizations of SVD that both go under many different names, I refer to them as PARAFAC and HOSVD.

First I will present some of the properties of the SVD for comparative purposes. Any $I_1 \times I_2$ matrix D, where it can be assumed for simplicity, $I_1 \geq I_2$, can be rewritten as the product of three matrices. That is, D is decomposed:

$$D = U\Sigma V^T.$$

U and V are orthogonal matrices of sizes $I_1 \times I_2$ and $I_2 \times I_2$ respectively and Σ, the singular matrix, is diagonal. By orthogonal I mean the columns of U and V, $\mathbf{u}_i$

and $\mathbf{v}_i$, satisfy $\mathbf{u}_i \cdot \mathbf{u}_j = \delta_{ij}$ and $\mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij}$ where δ_{ij} is the Kronecker delta. The diagonal elements of $\Sigma_{ii} = \sigma_i$ are all positive and sorted,

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m \geq 0$$

.

In order to elucidate the strength of this decomposition in interpreting data, it is helpful to rewrite it schematically.

$$D = [\mathbf{u}_1 \dots \mathbf{u}_n] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_n^T \end{bmatrix}$$

From this it becomes apparent that by applying an outer product expansion we get

$$D = \sum_{i=1}^{I_1} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

which shows that D can be rewritten as a sum of rank-1 matrices, each of which is independent and uncoupled from all other matrices in the sum. The significance of each matrix in the sum is determined by the singular values σ_i. In the case that the rank of D is k then $\sigma_i = 0$ for $i > k$. The Eckart-Young theorem also states that the best, in least square sense, low rank approximation of a matrix is the truncated sum[23].

This stems from the way in which U and V are determined. $\mathbf{u}_1$ ($\mathbf{v}_1^T$) captures the largest variance in the column (row) space of D, and the second column (row) $\mathbf{u}_2$ ($\mathbf{v}_2^T$) captures the remaining highest variance under the constraint that the vector is orthogonal to $\mathbf{u}_1$ ($\mathbf{v}_1$), and so on for all the columns (rows) (Fig. 4.1).

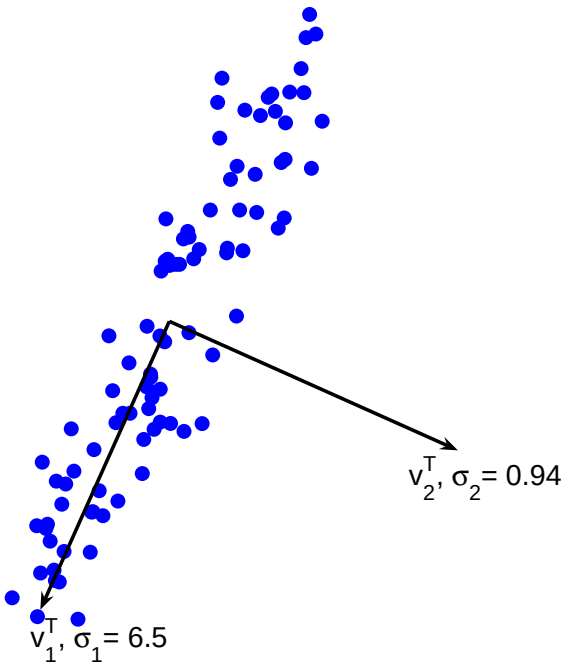


Figure 4.1: Example of scatter plot of $D \in \mathbb{R}^{100\times 2}$ and the two eigenvectors of V, v_1,v_2 ploted as black arrows. The most signficant eigenvector captures the largest variance as shown by the value of σ_1 and σ_2.

It has been shown that in cases where D represents the expression profiles of genes over multiple experiments, the columns of U, called eigenarrays, represent cellular states, and the rows of V, called eigengenes, contain corresponding biological processes[1]. The technique has also been shown to allow for selective filtering and reconstruction in subspaces to elucidate dynamics. The success of the SVD is what has motivated this thesis. In order to show how some of these properties carry over to higher-order data I will present some notations and definitions.

4.1 Background and Notations

The definition of a multidimensional array, henceforth referred to as a tensor, is simple. Any array of numbers, $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_m}$ is a tensor. The order of $\mathcal{T}$ is m and represents the number of modes in the tensor. A matrix is an order-two tensor and a vector is an order-one tensor. Each mode has a size or dimension. For example the jth dimension is I_j. Any element in the tensor is specified by $\mathcal{T}_{i_1 i_2 \ldots i_m}$. Subsets of the tensor are specified by ":", meaning all elements in the mode. For example, to extract the matrices or slabs out of a third order tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times I_m}$ we can specify $\mathcal{D}_{i::}$ which would extract the ith frontal slice. Likewise vectors in the three different directions can be specified by letting one of indices vary (e.g.,$\mathcal{D}_{i:k}$)

The literature is ambiguous about the use of the word rank. Sometimes the word rank is used interchangeably with the word order, but this generates confusion because the word rank has a clearly different meaning for matrices. I will use the definition that is most accepted in the multilinear-algebra literature, and is an obvious extension from linear algebra. The rank of tensor is the minimum number, F of vectors in a outer product required to recreate the tensor,

$$\mathcal{T} = \sum_{f=1}^{F} U_{1,:f} \otimes U_{2,:f} \otimes \ldots \otimes U_{n,:f}.$$

Unlike matrices where the rank is computable, the rank for a tensor is an NP-complete problem [30]. For data such as microarray data a few factors are needed to describe the majority of variance within the data, without being a strict rank reduction.

4.1.1 Tensor Multiplications

Before going any further I will define several types of tensor products. Let $\mathcal{A}$ and $\mathcal{B}$ be two tensors of equal size. Then the inner product between the tensors is

$$\mathcal{A} \cdot \mathcal{B} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_m=1}^{I_m} \mathcal{A}_{i_1 i_2 \ldots i_m} \mathcal{B}_{i_1 i_2 \ldots i_m}.$$

This leads to the natural definition of norm, $||\mathcal{A}||$, by extending the vector norm to the tensors known as Frobenius norm,

$$||\mathcal{A}||^2 = \mathcal{A} \cdot \mathcal{A}.$$

The matrix product or outerproduct, $\otimes$, generalizes to tensors as well:

$$\mathcal{T}_{i_1 i_2 \ldots i_{m+n}} = (\mathcal{A} \otimes \mathcal{B})_{i_1 i_2 \ldots i_{m+n}} = \mathcal{A}_{i_1 i_2 \ldots i_n} \mathcal{B}_{i_{n+1} 1 i_{n+2} \ldots i_{m+n}}$$

Flattening: representing tensors as matrices

Flattening is the action of creating matrix representation of a tensor and is extremely helpful in reformulating tensor operations in the more familiar matrix operations. The matrix representation has the column (row, ...) vectors of the tensor stacked one after each other. By being consistent in the ordering, an m-degree tensor can be unfolded into m different mode-matrices. The mode designated by a subscript indicates the vector direction of the tensor that become columns in the matrix (Fig. 4.2).

For example a third-order tensor $\mathcal{T} \in \mathbb{R}^{(I_1 \times I_2 \times I_3)}$ can be flattened into three different matrices, $T_1 \in \mathbb{R}^{I_1 \times (I_2 I_3)}$, $T_2 \in \mathbb{R}^{I_2 \times (I_3 I_1)}$ and $T_3 \in \mathbb{R}^{I_3 \times (I_1 I_3)}$, where the n-th index in the tensor is the row index of the corresponding T_n matrix and the column indices

are picked such that the first index varies more slowly than the second.

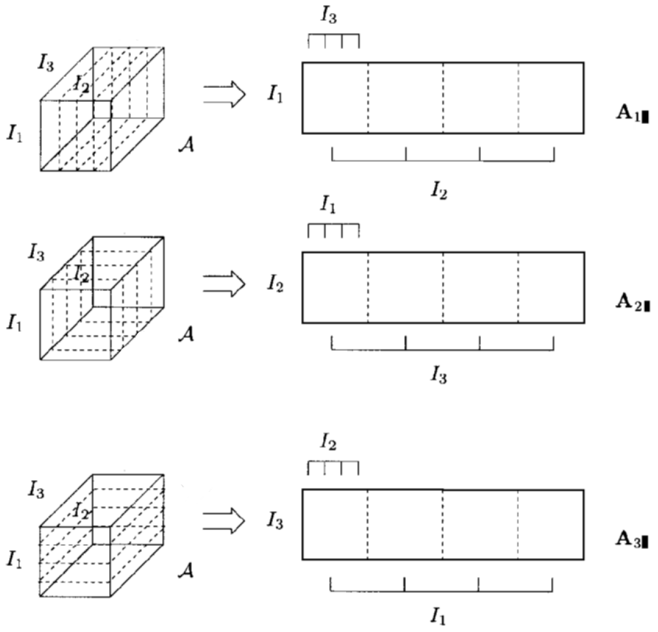


Figure 4.2: Unfolding of the third order tensor $\mathcal{A}$ of size $I_1 \times I_2 \times I_3$ into the three different modes: the $I_1 \times I_2 I_3$ sized A_1 matrix, the $I_2 \times I_3 I_1$ sized A_2 matrix and the $I_3 \times I_1 I_2$ sized A_3 matrix. Image reproduced from [18]

Tensor and matrix multiplication

The n-mode product, denoted by $\times_n$ between a tensor, $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_m)}$ and a matrix $U \in \mathbb{R}^{J_n \times I_n}$ is a special case of an inner product and is easily formulated in terms of the flattened tensor A_n,

$$T_n = UAn.$$

It can also be seen as a contraction over index n in the tensor and the second index of the matrix. (For example $\mathcal{A} \times_1 U = \sum_l U_{j_n l} \mathcal{A}_{l i_1 i_2 \ldots i_m}$).

4.2 Higher Order Singular Value Decomposition

The higher order singular value decomposition (HOSVD) was introduced by Tucker in the 1960's as a way to analyze psychometric data [62, 63] and has since been reformulated in terms of language familiar to linear algebra [18]. It is the latter formulation that I will present here. Let the m-order tensor $\mathcal{T}$, of size $I_1 \times I_2 \times \ldots \times I_m$, tabulate the measurement of a variable under $I_1 I_1 I_2 \ldots I_m$ different conditions such that every vector in any direction of the tensor only has one condition varying. The HOSVD is a transformation of $\mathcal{T}$,

$$\begin{aligned} \mathcal{T} &= \mathcal{R} \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_m U_m \\ \mathcal{T}_{klm\ldots} &= \sum_{i_1=1}^{I_2 I_3} \sum_{i_2=1}^{I_2} \ldots \sum_{i_m=1}^{I_m} \mathcal{R}_{i_1 i_2 \ldots i_m} U_{1,ki_1} U_{2,li_2} U_{3,mi_3} \ldots \end{aligned} \tag{4.1}$$

where $\times_1 U_1$, $\times_2 U_2$ and $\times_m U_m$ denote multiplications of the tensor $\mathcal{R}$ and the matrices U_1, U_2, and U_3 (Fig. 4.3). In this space the data tensor is represented by the core tensor $\mathcal{R}$, which in general, is full. The transformation matrices U_n defines the eigenvector basis set for each of the modes of $\mathcal{T}$ The vector in the i_1th column of U_1, $U_{1,:i_1}$, lists the signal of the i_1th eigenvector. Equivalently the transformation matrix U_n define the basis set for the nth mode of $\mathcal{T}$. Like the SVD the U_n matrices are orthonormal.

The multilinear HOSVD of Eq. 4.1 can be reformulated such that it decomposes the data tensor $\mathcal{T}$ into a linear superposition of $\leq (I_1 I_2 \ldots I_m)$ rank-1 subtensors, the superposition coefficients of which are the higher-order singular values, tabulated in the core tensor $\mathcal{R}$, that is,

$$\begin{aligned}\mathcal{T} &= \sum_{i_1=1}^{I_2 I_3} \sum_{i_2=1}^{I_2} \cdots \sum_{i_m=1}^{I_m} \mathcal{R}_{i_1 i_2 \ldots i_m} U_{1,:i_1} \otimes U_{2,:i_2} \otimes \ldots \otimes U_{m,:i_m} \\ &\equiv \sum_{i_1=1}^{I_2 I_3} \sum_{i_2=1}^{I_2} \cdots \sum_{i_m=1}^{I_m} \mathcal{R}_{i_1 i_2 \ldots i_m} \mathcal{S}(i_1, i_2, \ldots, i_m) \end{aligned} \tag{4.2}$$

where the subtensor $\mathcal{S}(i_1, i_2, \ldots, i_m)$ is the outer product, denoted by $\otimes$, of the i_1th eigenvector $U_{1,:i_1}$ and i_2th eigenvector $U_{2,:i_2}$ etc (Fig. 4.4). Following Eq. 4.2, we define the significance of a subtensor $\mathcal{S}(a, b, c)$ relative to all other subtensors in terms of the "fraction" $\mathcal{P}_{i_1 i_2 \ldots i_m}$,

$$\mathcal{P}_{i_1 i_2 \ldots i_m} = \frac{\mathcal{R}^2_{i_1 i_2 \ldots i_m}}{\sum_{i_1=1}^{I_2 I_3} \sum_{i_2=1}^{I_2} \cdots \sum_{i_m=1}^{I_m} \mathcal{R}^2_{i_1 i_2 \ldots i_m}},$$

which measures the fraction of the overall information in the data tensor that this subtensor captures. The "Shannon entropy" d,

$$0 \leq d = \frac{-1}{2\log(I_2 I_3)} \sum_{i_1=1}^{I_2 I_3} \sum_{i_2=1}^{I_2} \cdots \sum_{i_m=1}^{I_m} \mathcal{P}_{i_1 i_2 \ldots i_m} \log(\mathcal{P}_{i_1 i_2 \ldots i_m}) \leq 1, \tag{4.3}$$

measures the complexity of the data tensor from the distribution of the overall information among the different subtensors. This HOSVD holds for a tensor $\mathcal{T}$ of any order m. For a second-order tensor, that is, a matrix, this HOSVD reduces to the matrix SVD [27].

4.2.1 HOSVD Computation

I compute the transformation matrices U_n from the SVD of the n-mode flattened matrices $T_n = U_n D_n V_n^T$. The singular values, which are tabulated in the diagonal matrix D, are ordered in decreasing order, such that the eigenvectors, the column

vectors of U_n, are ordered in decreasing order of their relative significance in terms of the fraction of the overall information in the data tensor that each eigenarray captures (Figs. 5.1, 5.2, 5.3). For a real data tensor, the eigenvectors are unique up to phase factors of ± 1, such that each eigenvector captures both parallel and antiparallel data patterns, except in degenerate subspaces, defined by equal corresponding singular values in the diagonal matrices D_n. For example, the eigenvectors $U_{3,:i}$ and $U_{3,:j}$, which satisfy $D_{3,ii} \approx D_{3,jj}$, span an approximately degenerate subspace. We reformulate the HOSVD of Eqs. 4.1 and 4.2 with a unique orthogonal rotation of these two eigenarrays, which is selected by subjecting the rotated eigenvectors to a constraint, that may be advantageous in the interpretation and visualization of the data. We then compute the core tensor by multiplying the data tensor $\mathcal{T}$ and the transformation matrices U_1, U_2, and U_3, that is, $\mathcal{R} = \mathcal{T} \times_1 U_1^T \times_2 U_2^T \times_3 U_3^T$.

4.2.2 Approximately Degenerate Subtensor Space Rotation

We define a subset of subtensors as approximately degenerate if their corresponding higher-order singular values are approximately equal in magnitude and if $m-1$ of their m indices are equal, such that they are listed in a single vector in the core tensor $\mathcal{R}$. For example, the subtensors $\mathcal{S}(a,b,c)$ and $\mathcal{S}(k,b,c)$, which satisfy $\mathcal{R}_{abc} \approx \mathcal{R}_{kbc}$, span an "approximately degenerate subtensor space." We reformulate the HOSVD of Eq. 4.2 with a single rank-1 subtensor $\mathcal{S}(a+k,b,c)$ unique to the data tensor, which is composed of these two subtensors, with the corresponding higher-order singular value $\mathcal{R}_{a+k,b,c}$, that is,

$$\mathcal{R}_{abc}\mathcal{S}(a,b,c) + \mathcal{R}_{kbc}\mathcal{S}(k,b,c) = \mathcal{R}_{a+k,b,c}\mathcal{S}(a+k,b,c)$$

The subtensor $\mathcal{S}(a+k,b,c) \equiv U_{1,:,a+k} \otimes U_{2,:b} \otimes U_{3,:c}$ is computed from the outer product of $U_{1,:,a+k} \equiv \frac{\mathcal{R}_{abc}U_{1,:a} + \mathcal{R}_{kbc}U_{1,:k}}{\mathcal{R}_{a+k,b,c}}$, a normalized superposition of the eigenar-

rays $U_{1,:a}$ and $U_{1,:k}$, and the shared eigenvectors $U_{2,:b}$ and $U_{2,:c}$. This subtensor is unique to the data tensor, because it is defined by a unique rotation in the space spanned by $\mathcal{S}(a,b,c)$ and $\mathcal{S}(k,b,c)$.

4.3 Parallel Factorization (PARAFAC)

PARAFAC was invented simultaneously under two different names by two different groups; Harshmann in 1970 [29] called his decomposition PARAFAC while independently Carroll and Chang developed it under the name of CANDECOMP [13]. A data tensor $\mathcal{T}$ as defined above can be modeled by a sum of rank-1 tensors,

$$\mathcal{T} = \sum_{f=1}^{F} U_{1,:f} \otimes U_{2,:f} \otimes ... \otimes U_{m,:f} + \mathcal{E} \tag{4.4}$$

where the U_n are $I_n \times F$ sized real valued matrices, F is the number of factors in the model, and $\mathcal{E}$ is a tensor of residuals. The matrices U_n are chosen so as to minimize the square of these residuals.

The PARAFAC can be seen as a constrained version of HOSVD [32] in which, the core tensor $\mathcal{R}$ is superdiagonal with ones on the superdiagonal and 0 everywhere else, $\mathcal{R}_{i_1 i_2 ... i_m} = \delta_{i_1 i_2 ... i_m}$. This means that PARAFAC is more restrictive and will not have the same number of degrees of freedom as HOSVD. The freedom stems from the number of factors, F, chosen. PARAFAC has one advantage of HOSVD in that the decomposition is unique and not subject to rotational freedom [9]. That is, the factors in U_n can't be rotated without increasing the residuals in $\mathcal{E}$.

PARAFAC can be calculated using an alternating least square algorithm (ALS). ALS consists of dividing the parameters into several sets and then estimating each parameter set separately in a least square sense under the constraint of the other sets

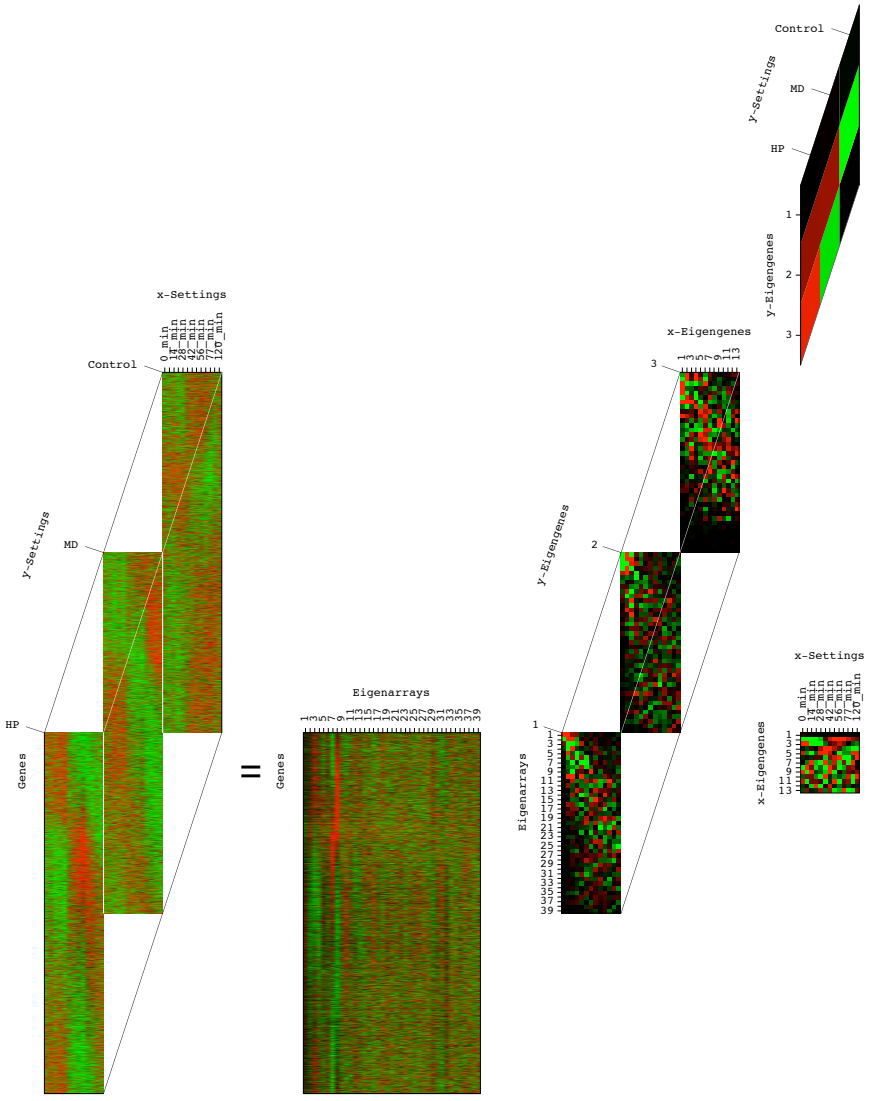


Figure 4.3: Higher-order singular value decomposition (HOSVD) is a transformation of the data tensor from the space of I_1-genes $\times I_2$-x-settings $\times I_3$-y-settings to the reduced space of $I_2 I_3 < I_1$-eigenarrays $\times$ I_2-x-eigengenes $\times$ I_3-y-eigengenes. Raster display of Eq. 4.1, $\mathcal{T} = \mathcal{R} \times_1 U^1 \times_2 U^2 \times_3 U^3$, using data presented in chapter 5, with overexpression (red), no change in expression (black), and underexpression (green). The expression of each array and eigenarray is centered at its gene-invariant level. The expression of each gene and x- and y-eigengene is centered at its x- and y-setting-invariant levels, respectively. The genes are sorted by the "angular distance" $\theta_{:} = \arctan(U^1_{:,8+2}/U^1_{:,3+7})$ between the two superpositions of eigenarrays $U^1_{:,8+2}$ and $U^1_{:,3+7}$, which define the expression variation across the genes in the ninth and tenth subtensors, respectively.

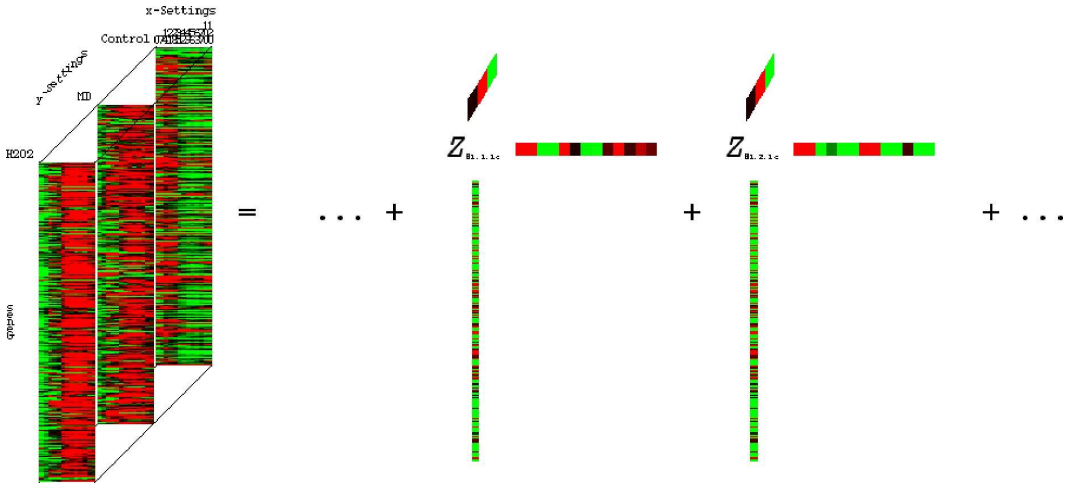


Figure 4.4: The HOSVD is reformulated such that it decomposes a data tensor into a linear superposition of all outerproducts of the eigenvectors $U_{n,:i_n}$, that is, rank-1 subtensors. Raster display of Eq. 4.2, $\mathcal{T} = \sum_{i_1=1}^{I_2 I_3} \sum_{i_2=1}^{I_2} \cdots \sum_{i_m=1}^{I_m} \mathcal{R}_{i_1 i_2 \ldots i_m} \mathcal{S}(i_1, i_2, \ldots, i_m)$, with overexpression (red), no change in expression (black) and underexpression (green).

in a round-robin manner. ALS allows nonlinear problems to be solved by solving multiple linear problems at each step. In so doing,ALS is a greedy algorithm [11] and is guaranteed to converge but not guaranteed to reach a global minima.

4.3.1 Calculating the PARAFAC using Alternating Least Square

To simplify the calculation of the PARAFAC decomposition it is helpful to reformulate the tensor decomposition in term of matrices. The Khatri-Rao matrix product defined, for two matrices U and V with the same number of columns, F is,

$$U|\otimes|V = [\mathbf{u_1} \otimes \mathbf{v_1}, \mathbf{u_2} \otimes \mathbf{v_2}, ..., \mathbf{u_F} \otimes \mathbf{v_F}]$$

The PARAFAC of the tensor $\mathcal{T}$ can then be expressed as matrix formula,

$$T_1 = U_1(U_m|\otimes|U_{m-1}|\otimes|\ldots|\otimes|U_2)^T.$$

To calculate the matrices U_n we follow the ALS algorithm as outlined [11]

1. Initiate the $U_n, n > 1$ matrices
2. $T_1 = U_1 Z^T$ where $Z = (U_m|\otimes|U_{m-1}|\otimes|\ldots|\otimes|U_2)$ which can be solved for U_1 in a least square sense by $U_1 = T_1(Z^T)^\dagger$.
3. $T_2 = U_2 Z^T$ where $Z = (U_1|\otimes|U_3|\otimes|\ldots|\otimes|U_m)$ which can be solved for U_2 in a least square sense by $U_2 = T_2(Z^T)^\dagger$.
4. ...repeat for all modes in m
5. Go to step 2 until changes in U_n is small.

There are a few shortcuts that can be done to avoid several of the matrix products and hence see minor speedups in practical applications. Contrary to my initial belief the initial values chosen for U_n have very little bearing on the convergence and speed of convergence. While exploring the data in chapter 5 I tried using the HOSVD eigenvectors as initial guesses considering that the factors found by PARAFAC are highly correlated with these but the gains compared to random initial matrices were unmeasurable.

4.3.2 How to Choose Number of Factors

A high number of components F will reduce the error in the decomposition but can result in over-fitting and multiple factors highly correlated with each other. While several empirical methods have been developed to explore the number of factors the core consistency offers a metric of success of a fit [10]. The idea is that a good model

will remain a valid model when restrictions of it are lifted. I mentioned earlier that PARAFAC can be considered a constrained version of HOSVD with a superdiagonal core tensor $\mathcal{I}$. When calculating the core consistency the factor matrices, U_n are calculated and then a full core tensor is fitted in a least square sense. If the model is good this new core tensor, $\mathcal{R}$, will remain close to superdiagonal. This can be reviewed as core consistency plots or a numerical representation is given by,

$$100\frac{\sum_{i=1}^{F}\sum_{j=1}^{F}\sum_{k=1}^{F}(\mathcal{I}_{ijk}-\mathcal{R}_{ijk})^2}{\sum_{i=1}^{F}\sum_{j=1}^{F}\sum_{k=1}^{F}\mathcal{I}_{ijk}}.$$

Chapter 5

A Case Study: Integrative Analysis of mRNA Expression from Yeast Cell Cycle Time Courses Under Different Oxidative Stress Conditions

[1] A single DNA microarray probes the genome-scale signal of I_1 genes of a cellular system in a single sample. A series of I_2 arrays probes I_2 different samples at I_2 different time points. A series of I_3 arrays probes the genome-scale signal under I_3 different conditions for each given time point. Let the third-order tensor $\mathcal{T}$, of size $I_1 \times I_2 \times I_3$, tabulate the genome-scale signal for all genes at all different times under all different conditions. Each element of $\mathcal{T}$, that is, $\mathcal{T}_{klm}$, is the signal

[1]The work presented in this chapter is a summary of results published in [45] and the data and mathematical details are available at http://www.bme.utexas.edu/research/orly/HOSVD.

measured for the kth gene at the lth time point and mth condition. Each column vector of $\mathcal{T}$, that is, $\mathcal{T}_{:lm}$, lists the genome-scale signal measured under the lth time point and mth condition. The time- and condition-row vectors, $\mathcal{T}_{k:m}$ and $\mathcal{T}_{kl:}$, list the signal measured for the kth gene under the mth condition across all time-points, and under the lth time-point across all conditions, respectively. Specifically I will present the results of a decomposition of a data tensor that tabulates relative mRNA expression levels of $I_1 = 4,329$ yeast Saccharomyces cerevisiae genes across $I_2 = 13$ time points sampled from each of $I_3 = 3$ cell cycle time courses of cultures synchronized by the pheromone α-factor. The three different cell cycle time courses are under different oxidative stress conditions: Exposures to (i) $\approx$0.2 mM hydrogen peroxide(HP), and (ii) $\approx$2 mM menadione(MD), starting at 25 min after 90 min of incubation in $\approx$7 nM α-factor, monitored by Shapira et al. [53] and (iii) a control time course, synchronized by 120 min of incubation in 7 nM α-factor, monitored by Spellman et al. [56]. The time points sample approximately two cell cycle periods in the control culture. The first period of 63 min is sampled at 7-min intervals. The second period is sampled at 77, 98, and 119 ± 2 min.

Each relative expression level is presumed valid when the signal-to-background ratio is 1.1 for both channels or the synchronized culture and asynchronous reference. The 4,329 genes have valid data in at least eight time points in each course, and at least 32 of the $I_2 I_3 = 39$ arrays. We use SVD to estimate the missing data in each time course separately assuming that 3 eigengenes represent the dynamics of each dataset (see section 3.3.2). Each array was normalized by its norm $||\mathcal{T}_{:lm}||$

5.1 Annotations of the Genes in the Data Tensor

Of the $4,329$ genes, the mRNA expression of 579 was traditionally or microarray-classified as cell cycle-regulated [56]. The expression of 312 and 680 genes was

microarray-classified as regulated by pheromone [51] or environmental stress [26], respectively. We annotate each of the genes as a DNA-binding target of either one of 19 transcription factors and four replication initiation proteins if the microarray-assigned P value for the binding of that protein to at least one of the probes that maps to that gene is < 0.02 [28, 67, 54]. The DNA-binding occupancy levels of the oxidative stress response activators and the pheromone response factors were measured after a 30-min exposure to 4 mM HP or 3 nM -factor, respectively. The cell cycle factors, Stb5 and the replication initiation proteins were measured at steady growth conditions (Fig. 5.6 and Table 5.1, 5.2).

5.2 HOSVD

The $N = 3$-mode SVD, a HOSVD of the third-order data tensor, is a transformation of the data tensor from the space of I_1-genes $\times I_2-$time-points $\times I_3-$conditions to the reduced space of $I_2 I_3 < I_1$-eigenarrays $\times I_2-$time-eigengenes $\times I_3-$condition-eigengenes,

$$\mathcal{T} = \mathcal{R} \times_1 U_1 \times_2 U_2 \times_3 U_3. \tag{5.1}$$

The transformation matrices U_1, U_2, and U_3 are calculated using the SVD of the three mode flattenings of $\mathcal{T}$ (Figs. 5.1–5.3). I rotate the approximately degenerate second and third condition-eigengenes, $U_{3,:2}$ and $U_{3,:3}$, such that the rotated $U_{3,:3}$ describes over- and underexpression in response to HP and MD, respectively, and steady-state expression in the control time course, that is, $U_{3,33} = 0$ (Fig. 5.4).

I then compute the core tensor, $\mathcal{R}$ by multiplying the data tensor $\mathcal{T}$ and the transformation matrices U_1, U_2, and U_3, that is, $\mathcal{R} = \mathcal{T} \times_1 U_1^T \times_2 U_2^T \times_3 U_3^T$ (Fig. 5.5). Degenerate subtensors - that is, tensors for which the corresponding higher-order singular values are approximateely equal in magnitude and where two out of three

mRNA Expression							
	1	2	3	4	5+2	8+2	3+7
Up	1	1	2.8×10^{-13}	1.2×10^{-15}	4.4×10^{-4}	1.2×10^{-1}	8.7×10^{-22}
Down	3.4×10^{-21}	5.1×10^{-13}	1	1	1	4.7×10^{-1}	1
Up	2.6×10^{-1}	4.5×10^{-13}	5.0×10^{-21}	3.9×10^{-19}	1.3×10^{-3}	3.5×10^{-3}	9.6×10^{-7}
Down	6.3×10^{-1}	1.4×10^{-4}	3.3×10^{-1}	6.3×10^{-1}	7.3×10^{-3}	1.7×10^{-2}	1.1×10^{-3}
G1	1.9×10^{-1}	9.1×10^{-8}	4.0×10^{-1}	5.3×10^{-1}	1.9×10^{-1}	4.0×10^{-1}	3.3×10^{-12}
G2/M	6.2×10^{-3}	3.3×10^{-2}	6.8×10^{-2}	6.8×10^{-2}	1.1×10^{-7}	8.6×10^{-4}	5.0×10^{-1}
M/G1	3.4×10^{-7}	3.8×10^{-1}	1.9×10^{-2}	2.2×10^{-3}	2.2×10^{-1}	2.2×10^{-1}	4.0×10^{-5}
Protein-DNA binding							
AFT2	9.6×10^{-1}	1	1.0×10^{-2}	2.5×10^{-1}	6.7×10^{-1}	9.0×10^{-1}	4.3×10^{-3}
CIN5	2.0×10^{-1}	5.9×10^{-1}	1.8×10^{-4}	1.8×10^{-4}	1.7×10^{-2}	3.1×10^{-1}	1.3×10^{-3}
MSN2	9.3×10^{-1}	9.3×10^{-1}	1.2×10^{-8}	8.9×10^{-5}	8.1×10^{-2}	2.2×10^{-2}	2.6×10^{-6}
MSN4	9.8×10^{-1}	9.8×10^{-1}	1.4×10^{-8}	6.4×10^{-4}	7.0×10^{-2}	7.0×10^{-2}	2.8×10^{-5}
SKN7	9.0×10^{-1}	6.2×10^{-1}	1.8×10^{-4}	2.5×10^{-3}	1.8×10^{-1}	8.3×10^{-1}	2.2×10^{-2}
YAP6	3.8×10^{-1}	9.4×10^{-1}	9.9×10^{-4}	1.3×10^{-2}	9.6×10^{-2}	5.3×10^{-2}	9.9×10^{-4}
YAP7	9.3×10^{-1}	1	6.5×10^{-6}	3.6×10^{-1}	1.2×10^{-1}	9.6×10^{-1}	1.3×10^{-2}
DIG1	7.0×10^{-1}	1.2×10^{-7}	6.3×10^{-7}	6.2×10^{-10}	1.8×10^{-2}	6.9×10^{-3}	4.5×10^{-2}
STE12	9.3×10^{-2}	7.8×10^{-8}	4.1×10^{-5}	9.0×10^{-12}	5.1×10^{-2}	9.3×10^{-2}	5.6×10^{-3}
TEC1	7.6×10^{-2}	1.4×10^{-6}	5.9×10^{-6}	1.4×10^{-6}	1.6×10^{-2}	1.6×10^{-2}	1.6×10^{-2}
MBP1	3.2×10^{-2}	5.7×10^{-4}	5.1×10^{-1}	7.6×10^{-1}	2.6×10^{-1}	3.8×10^{-1}	7.6×10^{-3}
SWI4	2.2×10^{-2}	2.1×10^{-5}	2.0×10^{-1}	1.2×10^{-1}	2.9×10^{-1}	4.0×10^{-1}	1.2×10^{-1}
SWI6	1.3×10^{-1}	4.6×10^{-5}	5.5×10^{-1}	2.0×10^{-1}	5.5×10^{-1}	7.5×10^{-2}	8.9×10^{-4}
FKH2	2.7×10^{-1}	7.7×10^{-1}	6.5×10^{-1}	5.1×10^{-1}	6.1×10^{-2}	1.7×10^{-2}	7.7×10^{-1}
NDD1	9.1×10^{-1}	1.4×10^{-1}	1.4×10^{-1}	4.3×10^{-2}	8.3×10^{-4}	4.8×10^{-3}	2.2×10^{-1}
MCM1	3.7×10^{-1}	5.6×10^{-3}	5.1×10^{-2}	5.1×10^{-2}	5.6×10^{-3}	1.6×10^{-1}	1.6×10^{-1}
ACE2	7.9×10^{-4}	2.1×10^{-1}	1.2×10^{-1}	9.0×10^{-1}	3.4×10^{-1}	9.7×10^{-1}	4.9×10^{-1}
SWI5	1.7×10^{-2}	5.9×10^{-1}	3.1×10^{-1}	1.7×10^{-2}	1.2×10^{-1}	7.3×10^{-1}	3.1×10^{-1}
STB5	8.9×10^{-1}	5.8×10^{-1}	2.4×10^{-1}	7.6×10^{-1}	3.0×10^{-2}	9.9×10^{-1}	6.6×10^{-2}
MCM3	9.3×10^{-1}	9.3×10^{-1}	1.4×10^{-4}	7.5×10^{-6}	1.3×10^{-1}	1.3×10^{-1}	3.7×10^{-3}
MCM4	1	8.9×10^{-1}	1.3×10^{-3}	2.8×10^{-3}	1.8×10^{-1}	1.8×10^{-1}	6.0×10^{-3}
MCM7	9.4×10^{-1}	9.4×10^{-1}	2.4×10^{-4}	1.4×10^{-3}	1.8×10^{-1}	1.8×10^{-1}	1.2×10^{-2}
ORC1	1	9.8×10^{-1}	1.1×10^{-2}	2.6×10^{-1}	6.8×10^{-2}	2.6×10^{-1}	2.3×10^{-3}

Table 5.1: Parallel associations by annotations of the eigenarrays and superpositions of eigenarrays that define expression variation across genes in all ten most significant subtensors.

indices are equal - $\mathcal{S}(4, 2+3, 1)$, $\mathcal{S}(5+2, 1, 3)$, $\mathcal{S}(8+2, 4, 3)$, and $\mathcal{S}(3+7, 2, 3)$ are rotated (see section 4.2.2).

5.2.1 Significant Subtensors Represent Independent Biological Programs or Experimental Phenomena

We find that significant subtensors represent independent biological programs or experimental phenomena common to all three studies or exclusive to either one or two of the studies, including the subtle differential effects of HP and MD on cell cycle

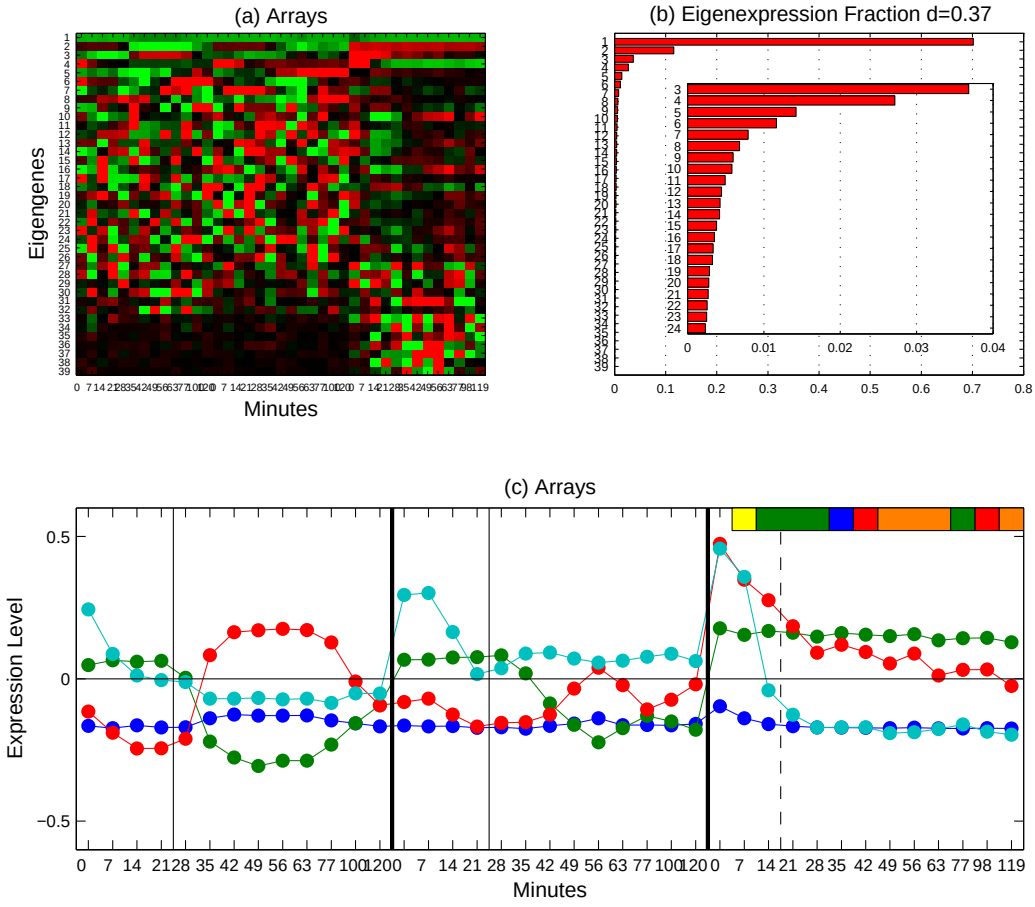


Figure 5.1: The eigengenes V_1^T that correspond to the eigenarrays U_1, which are computed from the SVD of the matrix $T_1 = (\mathcal{T}_{:11}, \ldots, \mathcal{T}_{:1I_2}, \ldots, \mathcal{T}_{:I_2I_3}) = U_1 D_1 V_1^T$. (a) Raster display of V^T, the expression of $I_2I_3 = 39$ eigengenes in 39 arrays, corresponding to 13 time points each in three cell cycle time courses, with overexpression (red), no change in expression (black), and underexpression (green) around the steady state of expression, which is captured by the first eigengene. (b) Bar chart of the corresponding fractions of eigenexpression. The entropy of the matrix T_1 is 0.37. (c) Line-joined graphs of the first (blue), second (green), third (red), and fourth (cyan) eigengenes. The time points in the control time course are color-coded according to their cell cycle classification: M/G1 (yellow), G1 (green), S (blue), S/G2 (red), and G2/M (orange). The grid lines mark the dissipation of the response to α-factor in the control time course (dashed) and the start of exposure to either HP or MD, at 20 and 25 min, respectively.

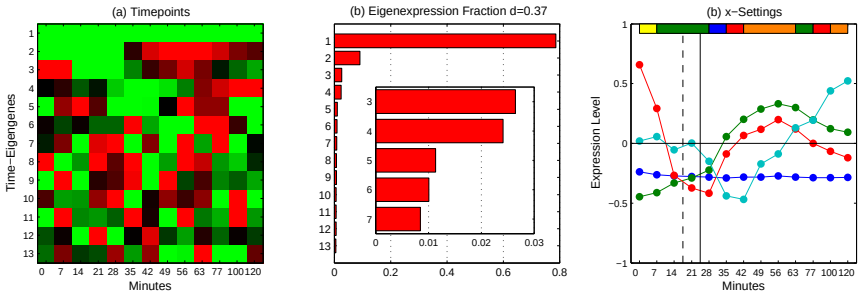


Figure 5.2: The time-eigengenes U_2 , which are computed from the SVD of the, $I_2 \times I_1 I_3$ sized, matrix $T_2 = U_2 D_2 V_2^T$. (a) Raster display of U_2^T, the expression of $I_2 = 13$ time-eigengenes in the 13 time points. (b) Bar chart of the corresponding fractions of eigenexpression. The entropy of the matrix T_2 is 0.37. (c) Line-joined graphs of the first (blue), second (green), third (red), and fourth (cyan) time-eigengenes. The time points are color-coded according to their cell cycle classification in the control time course: M/G1 (yellow), G1 (green), S (blue), S/G2 (red), and G2/M (orange). The grid lines mark the dissipation of the response to α-factor in the control time course (dashed) and the start of exposure to either HP or MD, at 20 and 25 min, respectively.

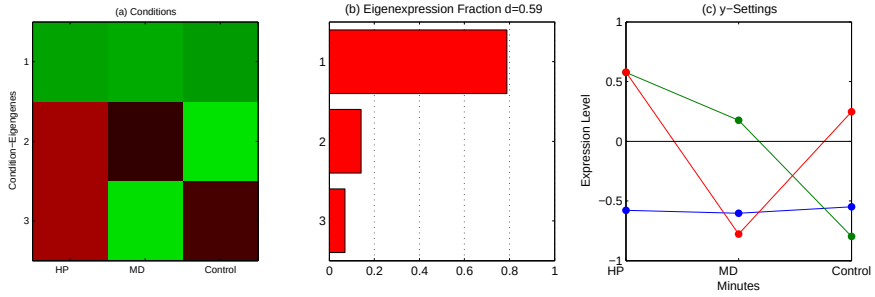


Figure 5.3: The condition-eigengenes U_3 , which are computed from the SVD of the, $I_3 \times I_1 I_2$ sized, matrix $T_3 = U_3 D_3 V_3^T$, before rotation of the approximately degenerate second and third condition-eigengenes, $U_{3,:2:}$ and $U_{3,:3}$. (a) Raster display of U_3^T, the expression of $I_3 = 3$ condition-eigengenes in the three oxidative stress conditions. (b) Bar chart of the corresponding fractions of eigenexpression. The entropy of the matrix T_3 is 0.59. (c) Line-joined graphs of the first (blue), second (green), and third (red) condition-eigengenes before rotation.

mRNA Expression							
	1	2	3	4	5+2	8+2	3+7
Up	2.1×10^{-43}	1.8×10^{-60}	5.8×10^{-1}	9.9×10^{-1}	8.7×10^{-22}	7.3×10^{-21}	1.9×10^{-1}
Down	1	1	1	1.3×10^{-2}	1	1	1
Up	9.0×10^{-11}	3.5×10^{-3}	8.3×10^{-1}	3.5×10^{-3}	1.5×10^{-1}	1.5×10^{-1}	5.5×10^{-1}
Down	3.3×10^{-1}	4.8×10^{-1}	3.5×10^{-2}	2.6×10^{-15}	2.1×10^{-1}	7.7×10^{-1}	3.3×10^{-1}
G1	2.8×10^{-1}	9.7×10^{-1}	1.2×10^{-1}	1.9×10^{-2}	2.8×10^{-1}	8.7×10^{-1}	9.4×10^{-1}
G2/M	3.3×10^{-2}	2.2×10^{-1}	6.6×10^{-1}	9.7×10^{-10}	8.0×10^{-1}	5.0×10^{-1}	5.0×10^{-1}
M/G1	2.2×10^{-3}	2.2×10^{-1}	3.8×10^{-1}	9.2×10^{-1}	1.1×10^{-1}	1.9×10^{-2}	1.1×10^{-1}
Protein-DNA binding							
AFT2	2.2×10^{-4}	7.3×10^{-5}	9.0×10^{-1}	1.5×10^{-1}	2.2×10^{-2}	2.5×10^{-1}	6.7×10^{-1}
CIN5	6.3×10^{-9}	1.1×10^{-7}	1.7×10^{-2}	1.2×10^{-1}	5.6×10^{-6}	1.1×10^{-7}	1.7×10^{-2}
MSN2	1.1×10^{-16}	1.5×10^{-17}	2.2×10^{-2}	8.1×10^{-2}	1.3×10^{-10}	2.9×10^{-5}	1.4×10^{-1}
MSN4	2.0×10^{-16}	2.2×10^{-23}	7.0×10^{-2}	4.3×10^{-1}	7.5×10^{-10}	7.7×10^{-7}	1.6×10^{-3}
SKN7	9.4×10^{-11}	8.0×10^{-8}	7.0×10^{-2}	1.2×10^{-1}	8.5×10^{-6}	1.1×10^{-2}	1.1×10^{-2}
YAP6	2.5×10^{-3}	2.5×10^{-3}	2.6×10^{-1}	3.8×10^{-1}	5.3×10^{-2}	2.5×10^{-3}	9.6×10^{-2}
YAP7	1.7×10^{-9}	3.0×10^{-11}	7.0×10^{-1}	7.5×10^{-2}	2.2×10^{-6}	1.7×10^{-9}	7.5×10^{-2}
DIG1	7.3×10^{-4}	5.1×10^{-1}	8.6×10^{-1}	2.1×10^{-4}	4.5×10^{-2}	8.6×10^{-1}	8.6×10^{-1}
STE12	5.1×10^{-2}	9.4×10^{-1}	1.6×10^{-1}	3.5×10^{-4}	1.6×10^{-1}	9.8×10^{-1}	5.1×10^{-1}
TEC1	6.5×10^{-3}	9.4×10^{-1}	7.3×10^{-1}	2.7×10^{-4}	1.5×10^{-1}	8.6×10^{-1}	3.9×10^{-1}
MBP1	3.8×10^{-1}	7.6×10^{-1}	8.6×10^{-1}	7.6×10^{-3}	6.4×10^{-1}	6.4×10^{-1}	5.1×10^{-1}
SWI4	2.0×10^{-1}	5.3×10^{-1}	2.0×10^{-1}	2.3×10^{-6}	9.3×10^{-1}	8.6×10^{-1}	2.2×10^{-2}
SWI6	7.5×10^{-2}	4.2×10^{-1}	2.0×10^{-1}	8.9×10^{-4}	8.8×10^{-1}	5.5×10^{-1}	4.9×10^{-3}
FKH2	3.8×10^{-1}	7.7×10^{-1}	3.8×10^{-1}	6.1×10^{-4}	7.7×10^{-1}	9.3×10^{-1}	1.7×10^{-2}
NDD1	1.4×10^{-1}	7.9×10^{-2}	7.2×10^{-1}	8.3×10^{-4}	5.9×10^{-1}	4.5×10^{-1}	4.8×10^{-3}
MCM1	5.1×10^{-2}	6.5×10^{-1}	5.1×10^{-2}	9.3×10^{-2}	6.5×10^{-1}	5.1×10^{-2}	1.6×10^{-1}
ACE2	6.6×10^{-1}	1.2×10^{-1}	6.6×10^{-1}	2.1×10^{-1}	1.2×10^{-1}	6.6×10^{-1}	2.1×10^{-1}
SWI5	2.0×10^{-1}	1.2×10^{-1}	7.3×10^{-1}	7.3×10^{-1}	4.5×10^{-1}	4.5×10^{-1}	7.9×10^{-3}
STB5	2.4×10^{-1}	1.3×10^{-1}	4.0×10^{-1}	9.7×10^{-1}	7.6×10^{-1}	4.6×10^{-4}	8.9×10^{-1}
MCM3	2.1×10^{-5}	3.3×10^{-4}	3.3×10^{-4}	1.9×10^{-1}	1.7×10^{-3}	1.4×10^{-2}	4.7×10^{-2}
MCM4	5.5×10^{-4}	3.4×10^{-5}	1.2×10^{-2}	1.8×10^{-1}	2.8×10^{-3}	2.3×10^{-2}	1.2×10^{-1}
MCM7	2.4×10^{-4}	5.8×10^{-4}	1.2×10^{-2}	1.2×10^{-1}	4.3×10^{-2}	3.0×10^{-3}	1.8×10^{-1}
ORC1	1.1×10^{-2}	1.1×10^{-2}	2.6×10^{-1}	6.8×10^{-2}	6.8×10^{-2}	6.8×10^{-2}	2.6×10^{-1}

Table 5.2: Antiparallel associations by annotations of the eigenarrays and superpositions of eigenarrays that define expression variation across genes in all ten most significant subtensors.

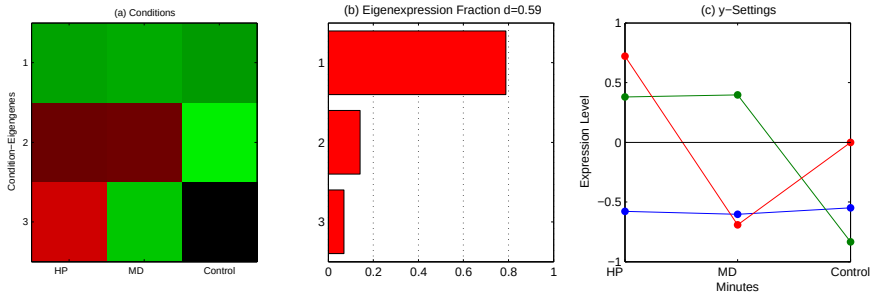


Figure 5.4: The condition-eigengenes U_3 after rotation of the approximately degenerate second and third condition-eigengenes, $U_{3,:2}$ and $U_{3,:3}$, under the constraint that the expression of the rotated third y-eigengene in the control time course is at steady state, that is, $U_{3,33} = 0$. (a) Raster display of U_3^T. (b) Bar chart of the fractions of the condition-eigengenes. (c) Line-joined graphs of the first condition-eigengene (blue) and the second (green) and third (red) rotated condition-eigengenes. The rotated $U_{3,:2}$ describes overexpression in response to HP and MD, and underexpression in the control time course. The rotated $U_{3,:3}$ describes over- and underexpression in response to HP and MD, respectively, and steady-state expression in the control time course.

progression. We also find that this subtensor interpretation is robust to variations in the data selection cutoffs.

Steady state

The first and most significant subtensor $\mathcal{S}(1,1,1)$ captures $\mathcal{P}_{111} \approx 70\%$ of the overall expression information in the data tensor. The corresponding higher-order singular value is $\mathcal{R}_{111} < 0$ (Fig. 5.5a). Following the P values for the distribution of the genes among each of the subsets of $k = 200$ genes with largest and smallest levels of expression in the first eigenarray $U_{1,:1}$, which defines the expression variation across the genes in this subtensor, this eigenarray is antiparallel-associated with mRNA expression in response to environmental stress and the pheromone, and is parallel-associated with overexpression during the cell cycle stage M/G1 (Fig. 5.6 and Tables 5.1, 5.2). Consistently, this eigenarray is also antiparallel-associated with the ex-

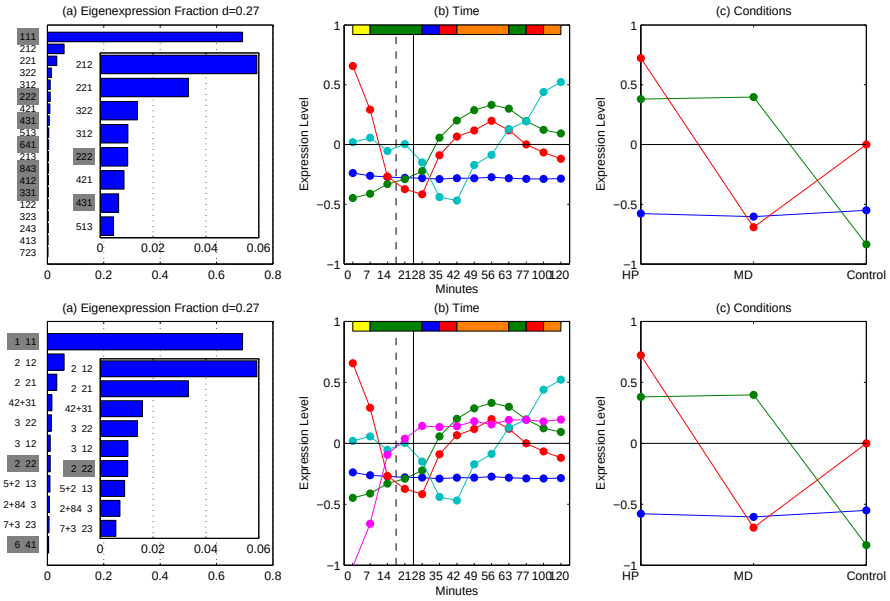


Figure 5.5: Significant HOSVD subtensors - before rotation (top row) and after rotation (bottom row) of the approximately degenerate subtensor spaces $\mathcal{S}(4, 2+3, 1)$, $\mathcal{S}(5+2, 1, 3)$, $\mathcal{S}(8+2, 4, 3)$, and $\mathcal{S}(3+7, 2, 3)$. (a) Bar chart of the fractions of the most significant subtensors. The higher-order singular values corresponding to subtensors highlighted in gray are < 0. The entropy of the data tensor is 0.27. (b) Line-joined graphs of the first (blue), second (green), third (red), and fourth (cyan) time-eigengenes and the superposition of the second and third time-eigengenes(magenta), which define the expression variation across time in these subtensors. The time points are color-coded according to their cell cycle classification in the control time course: M/G1 (yellow), G1 (green), S (blue), S/G2 (red), and G2/M (orange). The grid lines mark the dissipation of the response to α-factor in the control time course (dashed) and the start of exposure to either HP or MD, at 20 and 25 min, respectively. (c) Line-joined graphs of the first condition-eigengene (blue), and the second (green) and third (red) rotated condition-eigengenes, which define the expression variation across the oxidative stress conditions.

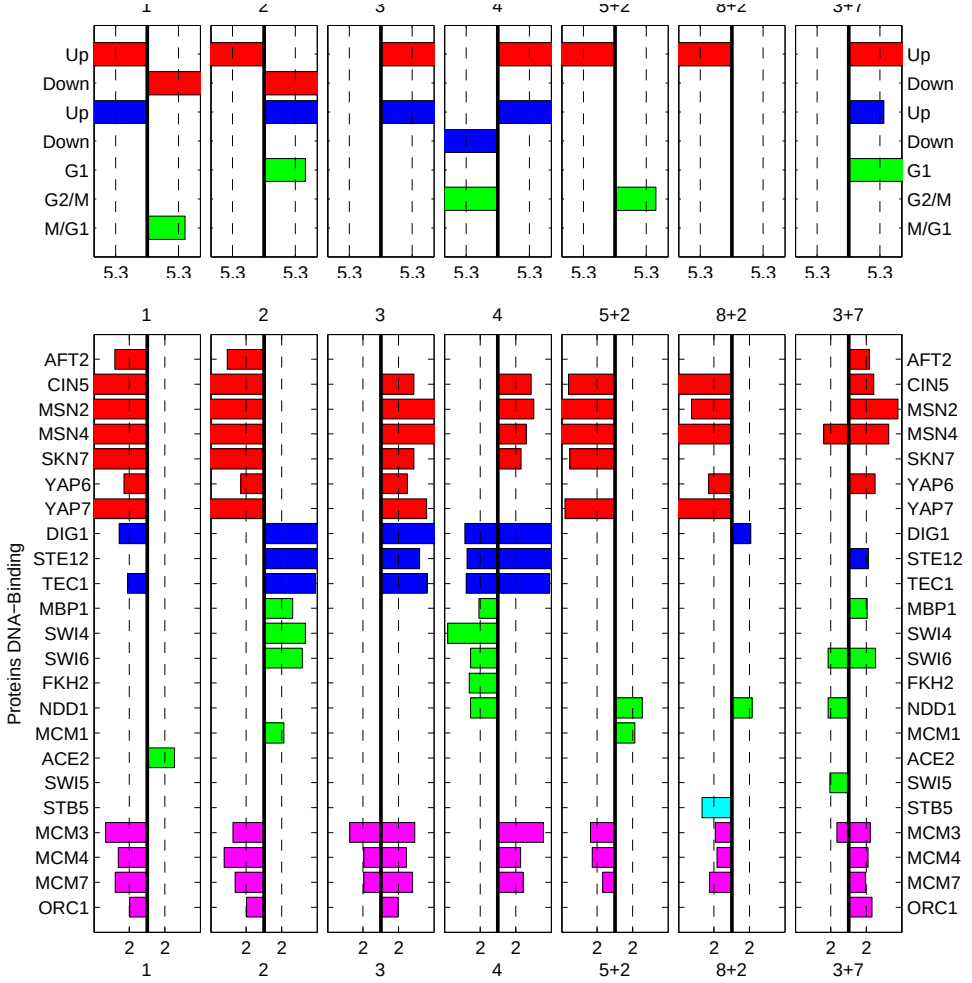


Figure 5.6: Associations by annotations of the eigenarrays and superpositions of eigenarrays that define expression variation across genes in all ten most significant subtensors. Bar chart of -$\log_{10}(P\text{value})$ for parallel (Right) and antiparallel (Left) enrichments of genes, which are expressed in response to environmental stress (red) or the pheromone (blue) or during the cell cycle (green), or of genes that are binding targets of oxidative stress activators (red), pheromone response (blue), or cell cycle (green) transcription factors, Stb5 (cyan) or replication initiation proteins (magenta).

pression of genes bound by oxidative stress response activators and the pheromone response factors Dig1 and Tec1, and is parallel-associated with the expression of genes bound by the M/G1 factor Ace2. The first time-eigengene $U_{2,:1}$, which defines the expression variation across time in this subtensor, describes time-invariant under-expression (Fig. 5.5b). The first condition-eigengene $U_{3,:1}$, which defines the expression variation across the oxidative stress conditions, describes condition-invariant under-expression (Fig. 5.5c). Taken together, the first subtensor is inferred to represent the steady state of mRNA expression in response to HP, MD, or α-factor, averaged over time and conditions.

Oxidative stress responses

The second, third, and seventh subtensors, $\mathcal{S}(2,1,2)$, $\mathcal{S}(2,2,1)$, and $\mathcal{S}(2,2,2)$, capture $\approx 6\%, 3.3\%$, and 1% of the overall information, respectively, with $\mathcal{R}_{212}, \mathcal{R}_{221} > 0$ and $\mathcal{R}_{222} < 0$. The second eigenarray is antiparallel-associated with expression in response to environmental stress and is parallel-associated with pheromone response and G1. The second time-eigengene describes a transition from under- to over-expression at 35 min. The second condition-eigengene describes over-expression in the HP- and MD-treated cultures and under-expression in the control culture. These subtensors are inferred to represent expression in response to oxidative stress: The second subtensor represents time-averaged response to the oxidative stress induced by HP and MD vs. the time-averaged response induced by α-factor. The third subtensor represents condition averaged expression variation across time in response to HP or MD exposure starting at 25 min, or in response to α-factor, which in the control culture dissipates at ≈ 20 min. The seventh subtensor represents oxidative stress response that varies across both time and conditions.

Pheromone responses

The fourth, fifth, and sixth subtensors, $\mathcal{S}(4, 2+3, 1), \mathcal{S}(3, 2, 2)$, and $\mathcal{S}(3, 1, 2)$, capture $\approx 1.6\%, 1.4\%$, and 1% of the overall information, with $\mathcal{R}_{4,2+3,1}, \mathcal{R}_{322}$, and $\mathcal{R}_{312} > 0$. The superposition of the second and third time-eigengenes describes an inverse time-decaying transition from over- to under-expression at 20 min. Both third and fourth eigenarrays are parallel-associated with expression in response to environmental stress and the pheromone. These subtensors are inferred to represent pheromone and pheromone- induced oxidative stress responses: The fourth subtensor represents a condition-averaged, time-decaying response. The fifth subtensor represents an α-factor response that varies across time and conditions. The sixth subtensor represents a time-averaged response to the α-factor in the HP- and MD-treated cultures vs. that in the control culture.

HP- vs. MD-Induced Expression

The eighth, ninth, and tenth subtensors, $\mathcal{S}(5{+}2, 1, 3)$, $\mathcal{S}(8{+}2, 4, 3)$, and $\mathcal{S}(3{+}7, 2, 3)$, capture $\approx 0.9\%$, 0.75%, and 0.6% of the overall information, with the corresponding higher-order singular values > 0. Of the corresponding superpositions of eigenarrays, $U_{1,:5+2}$ and $U_{1,:8+2}$ are antiparallel- and $U_{1,:3+7}$ is parallel-associated with expression in response to environmental stress and of oxidative stress activator-bound genes. Also, $U_{1,:5+2}$ and $U_{1,:8+2}$ are parallel- and $U_{1,:3+7}$ is antiparallel-associated with expression activated by the G2/M factor Ndd1. These subtensors are inferred to represent responses to the HP- vs. MD-induced oxidative stress: The eighth subtensor represents time-averaged under-expression. The ninth and tenth subtensors represent over-expression, starting at 25 and 35 min and peaking at 40 and 55 min, when the control culture is at S/G2 and G2/M, respectively (Fig. 5.7a). Taken together, oxidative stress-induced and G1 genes are over- and G2/M genes

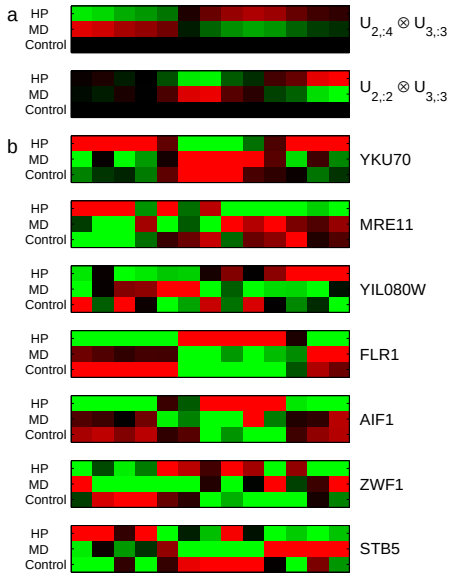


Figure 5.7: Eigengenes and genes that are significant in the HP vs. MD-induced responses. (a) Raster display of the outer products of the fourth and second time-eigengenes with the third condition-eigengene, $U_{2,:4} \otimes U_{3,:3}$ and $U_{2,:2} \otimes U_{3,:3}$, which define the expression variations across time and oxidative stress conditions in the ninth and tenth subtensors, $\mathcal{S}(8+2,4,3)$ and $\mathcal{S}(3+7,2,3)$, respectively. (b) Raster display of the expression of significant genes centered at the time and condition-invariant expression levels of each gene.

are under-expressed in the HP- vs. the MD-treated time course. These results are in agreement with the current understanding of the differences in the response to HP vs. the response to MD: The HP-treated culture arrests in G2/M after extended G1 and S stages in a manner that depends on inactivation of the Mcm1-Fkh2-Ndd1 transcription regulatory complex [53] and the DNA damage-induced RAD9 checkpoint, whereas the MD-treated culture continues through G2/M and M/G1 and arrests in G1 because of under-expression of the G1 cyclin-encoding CLN1 and CLN2 [25].

The eighth, ninth, and tenth subtensors classify the yeast genes according to the time dependence of their differential expression and identify the subsets of genes with largest and smallest expression in each subtensor as significant in the HP- vs. MD-induced responses in terms of the fraction of the information in either subtensor that they capture. The genome-scale picture that emerges from this data-driven analysis suggests that the evolutionarily highly conserved genes YKU70, MRE11, AIF1, and ZWF1, and the processes of retrotransposition, apoptosis, and the oxidative pentose phosphate pathway, that they are involved in, may play significant, yet previously unrecognized, roles in the difference between the effects of HP and MD on cell cycle progression in yeast.

Retrotransposition. Over-expression in the eighth and ninth subtensor and under-expression in tenth subtensors define genes of which time-averaged expression is greater in the MD- than the HP-treated culture and is modulated by a peak in the MD- and a trough in the HP-treated culture at $\approx$ 50 min, when the control culture is at G2/M. The most significant of these genes in terms of the fraction of the information in the eighth, ninth, and tenth subtensors that it captures is the yeast Ku protein-encoding YKU70 (Fig. 5.7b). Yku70 is a telomere maintenance protein, which is necessary for escape from the RAD9 checkpoint arrest in G2/M. In this process, Yku70 and the meiotic recombination protein Mre11 play antagonistic

roles, even though deletion of YKU70 is similar to that of MRE11 in its effect on nonhomologous end joining of DNA double-strand breaks [38]. Yku70 was shown to potentiate retrotransposition [22], whereas disruption of MRE11 was shown to increases retrotransposition levels [52]. We find MRE11 the 40th most significant gene with under-expression in the eighth, ninth and tenth subtensors. Consistently, the subset of the 200 most significant genes, which are anticorrelated with MRE11 in these subtensors, includes 16 of the 20 retrotransposon nucleocapsid genes in this data tensor, such as YIL080W, an enrichment that corresponds to a P value of $\approx 10^{-18}$.

Apoptosis. Among genes anticorrelated with YKU70 in the eighth, ninth, and tenth subtensors, the second most significant gene is FLR1, a multidrug transporter. This differential expression of FLR1 is consistent with the observation that its transcription is regulated by the oxidative stress factor YAP1 and is induced by HP but not by MD [43]. The 19th most significant gene is AIF1, which encodes the yeast apoptosis-inducing factor. Over-expression of AIF1, which with SKN7, SNQ2, and YAP1, constitutes the gene ontology "response to singlet oxygen" core [16], stimulates HP-induced apoptopic cell death [66]. This differential expression of AIF1 is consistent with the inactivation of the frog Xenopus laevis Ku70 during apoptosis [37].

Oxidative pentose phosphate pathway. Among genes correlated with AIF1 and anticorrelated with YKU70, the 18th most significant is ZWF1, which encodes the yeast glucose-6-phosphate dehydrogenase. Glucose-6-phosphate dehydrogenase catalyzes the first step of the pentose phosphate pathway, that is, the oxidative utilization of glucose, and is involved in response to HP. ZWF1 is among the 200 genes with the highest expression in the ninth subtensor, together with GND1 and SOL3, the two other genes in the gene ontology "oxidative brunch of the pentose-phosphate shunt" core in this data tensor, and STB5, an S/G2 gene that encodes

a transcription factor required for the regulation of the pentose phosphate pathway [36]. Consistently, the ninth subtensor is antiparallel-associated with expression of Stb5-bound genes (Fig. 5.6 and Tables 5.1, 5.2).

Oxidative Stress Response Is Correlated with Over-expression of Binding Targets of Replication Initiation Proteins.

Recently, a genome-scale correlation between the DNA binding of the replication initiation proteins Mcm3, Mcm4, and Mcm7 and under-expression of adjacent genes during G1 was discovered [1]. Replication initiation requires G1 binding of these proteins, which are involved in transcriptional silencing [42], at replication origins [20]. Therefore, we suggested that this correlation might be explained by a previously unknown mechanism of regulation. Now we uncover independently an equivalent genome-scale correlation: In all ten most significant subtensors and the corresponding seven eigenarrays and superpositions of eigenarrays, over-expression of binding targets of Mcm3, Mcm4, and Mcm7 correlates with expression in response to environmental stress and with over-expression of oxidative stress activator-bound genes. DNA damage as caused by oxidative stress is known to inhibit binding of origins by targeted degradation of the essential prereplicative complex protein Cdc6 [15, 8]. Taken together, we find that over-expression of binding targets of replication initiation proteins correlates with reduced, or even inhibited, binding of the origins. This correlation is in agreement with the recent observation that reduced efficiency of activation of origins correlates with local transcription [21, 55].

As with the correlation between the DNA binding of Mcm3, Mcm4, and Mcm7 and under-expression of adjacent genes during G1, this equivalent correlation between over-expression of binding targets of Mcm3, Mcm4, and Mcm7 and expression in response to stress may be due to either one of at least two mechanisms of regulation:

Stress-induced transcription of genes that are located near origins [50, 12] may reduce the binding efficiency of the adjacent origins. Or, reduced or even inhibited binding of origins by replication initiation proteins caused by degradation of Cdc6 may release genes that are located near origins for transcription. For example, the promoter region of the stress-induced FLR1, which includes Cin5 and Yap7 binding sites, overlaps with the yeast autonomously replicating sequence ARS209, and the stress-induced ZWF1 is transcribed in the direction of ARS1412 [14].

5.3 PARAFAC

The F component PARAFAC decomposition of the third-order data tensor of I_1-genes $\times I_2-$time-points $\times I_3-$conditions, is an approximate decomposition to the sum of F rank-1 tensors,

$$\mathcal{T} \approx \sum_{f=1}^{F} U_{1,:f} \otimes U_{2,:f} \otimes U_{3,:f} \tag{5.2}$$

$$\mathcal{T}_{ijk} \approx \sum_{f=1}^{F} U_{1,if} U_{2,jf} U_{3,kf}. \tag{5.3}$$

A high number of components F will reduce the error in the decomposition but can result in over-fitting. Core consistency diagnostics was developed for the purpose of determining the number of factors[10]. The equivalent of the core tensor of HOSVD in PARAFAC is a superdiagonal tensor, $\mathcal{I}$ with 1s on the superdiagonal and 0s everywhere else, which is implicitly included in the equations above. The core consistency is determined by using the U_1, U_2 and U_3 factors as determined by the PARAFAC and regressing a core tensor $\mathcal{R}$ that allows for off-diagonal elements

and comparing,

$$100\frac{(\sum_{i=1}^{F}\sum_{j=1}^{F}\sum_{k=1}^{F}(\mathcal{I}_{ijk}-\mathcal{R}_{ijk})^2}{\sum_{i=1}^{F}\sum_{j=1}^{F}\sum_{k=1}^{F}\mathcal{I}_{ijk}}.$$

The loading matrices U_1, U_2, and U_3 are calculated using an alternating least square approach (section 4.3.1). Multiple values of F yield the best model (Fig. 5.9) for $F = 2$ (Fig. 5.8).

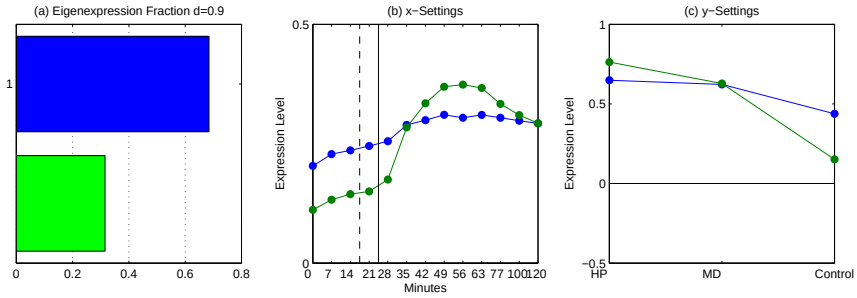


Figure 5.8: Two-component PARAFAC decomposition of data tensor $\mathcal{T}$ (a) Bar chart of the fractions of the most significant subtensors. (b) Line-joined graphs of the first (blue), second (green) time-factors, which define the expression variation across time in these subtensors. The grid lines mark the dissipation of the response to α-factor in the control time course (dashed) and the start of exposure to either HP or MD, at 20 and 25 min, respectively. (c) Line-joined graphs of the first condition-factor (blue), and the second (green), which define the expression variation across the oxidative stress conditions.

5.3.1 PARAFAC subtensors capture subset of HOSVD subtensors

We find that subtensor represent independent biological programs or experimental phenomena common to all three studies. The subtensor only capture a subset of the subtensors extracted by the HOSVD.

The first and most significant subtensor capture $\approx 65\%$ of the overall expression information in the data tensor. Following the P values for the distribution of the genes among each of the subsets of $k = 200$ genes with largest and smallest levels of

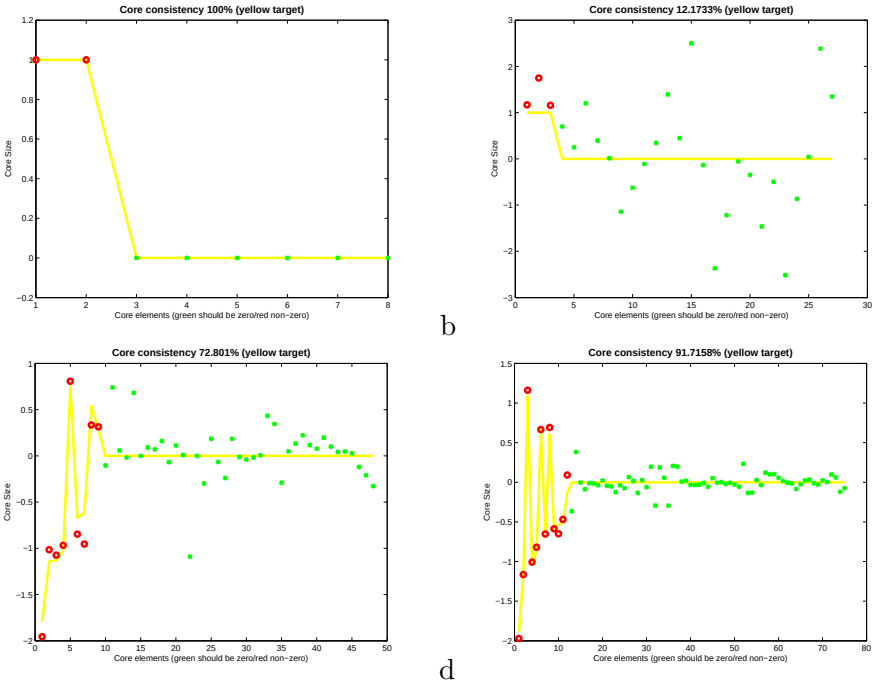


Figure 5.9: Core consistency plot of (a) two- (b) three- (c) four- (d) five-component PARAFAC decomposition. In the plots the red circles represent the superdiagonal elements and should preferable be non-zero while the green dots are the off superdiagonal elements that should be zero for a good model. The data tensor $\mathcal{T}$ is best represented by a two-component model.

expression in the first array-factor $U_{1,:1}$, which defines the expression variation across the genes in this subtensor, this array-factor is antiparallel-associated with mRNA expression in response to environmental stress and over-expression during the cell cycle stage M/G1 and is parallel-associated with mRNA expression in response to pheromone (Fig. 5.10). Consistently, this eigenarray is also antiparallel-associated with the expression of genes bound by oxidative stress response activators and the pheromone response factors Dig1 and Tec1 is parallel-associated. The first time-eigengene $U_{2,:1}$, which defines the expression variation across time in this subtensor, is steady state. Together this subtensor has a similar interpretation to $\mathcal{S}(1,1,1)$.

The second subtensor capture $\approx 35\%$ of the overall information. The array-factor is antiparallel-associated with expression in response to pheromone and G1 and parallel-associated with stress response. The second time-factor describes a transition from under- to over-expression at 35 min. The second condition-eigengene describes over-expression in the HP- and MD-treated cultures and slightly lower over-expression in the control culture. From this I infer that the second subtensor, like $\mathcal{S}(2,1,2)$, $\mathcal{S}(2,2,1)$, and $\mathcal{S}(2,2,2)$, represent expression in response to oxidative stress. Variation across time in response to HP or MD exposure starting at 25 min, or in response to α-factor, which in the control culture dissipates at ≈ 20 min and variation across conditions is captured.

5.4 Conclusions

We have shown that multilinear generalizations to SVD provide and integrative framework for analysis of DNA microarray data from different studies, where significant subtensors represent independent biological programs or experimental phenomena. The HOSVD, reformulated to decompose a data tensor into a linear superposition of rank-1 subtensors, especially so.

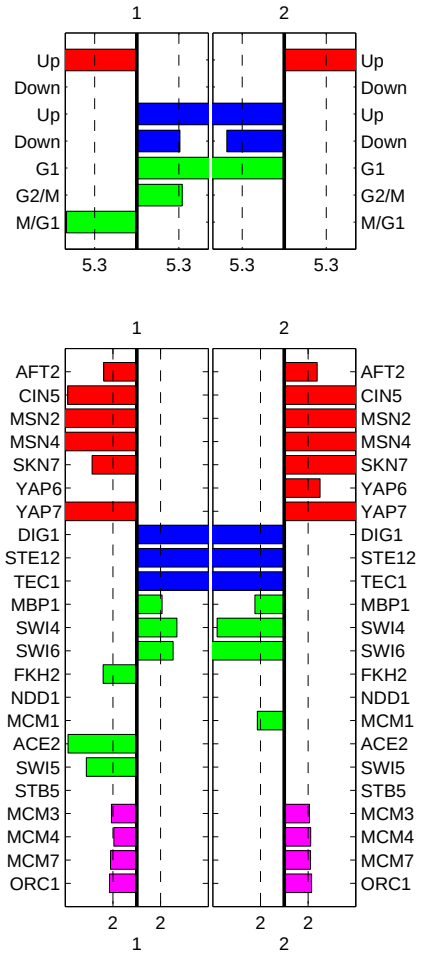


Figure 5.10: Associations by annotations of the array factors that define expression variation across genes the two subtensors. Bar chart of -$\log_{10}(P\text{value})$ for parallel (Right) and antiparallel (Left) enrichments of genes, which are expressed in response to environmental stress (red) or the pheromone (blue) or during the cell cycle (green), or of genes that are binding targets of oxidative stress activators (red), pheromone response (blue), or cell cycle (green) transcription factors, Stb5 (cyan) or replication initiation proteins (magenta).

By using this HOSVD in an integration of genome-scale mRNA expression data from three yeast cell cycle time courses, two of which are exposed to either HP or MD, we were able to find that the conserved genes YKU70, MRE11, AIF1, and ZWF1, and the processes of retrotransposition, apoptosis, and the oxidative pentose phosphate pathway that these genes are involved in, may play significant, yet previously unrecognized, roles in the differential effects of HP and MD on cell cycle progression. A genome-scale correlation between DNA replication initiation and RNA transcription, which is equivalent to a recently discovered correlation and might be due to a previously unknown mechanism of regulation, has been independently uncovered.[2]

[2]This work was supported by National Human Genome Research Institute Grant HG004302 (to Orly Alter)

Appendix A

Manuals to Tools for Analysis

A.1 enrichcommand help

NAME

enrichcommand - calculates enrichment of annotations.

SYNOPSIS

enrichcommand [opts] genesFile numbercols annotations1 [annotations2 ...]

DESCRIPTION

enrichcommand is a computational and visualization tool for enrichments of annotations among groups of genes. Utilizing a series of libraries for tracking mappings of labels to annotations, visualization and computation it manages to generate multiple types of reports from easy searchable text documents to detailed html documents including sparklines to publish ready figures. All

demonstrating the enrichment in terms of statistical enrichment using the hypergeometric distribution to obtain a p-value.

REQUIRED PARAMETERS

genesFile is a tab delimited file with gene names in the first column and each subsequent column representing a different group of these genes. Each gene has a number in all of the columns representing the strength of presence in the group. This could be a simple Preclustering file. In which case the highest expressed genes in each array will be examined.

numbercols A number indicating the number of columns in the allGenesFile that are labels. In the case that the first column of the file contains the genenames and the second column represents the first group then numbercols should be 1.

annotationsFiles One or more tab delimited files with two columns the first containing gene names and the second the annotation.

OPTIONS

--s value The number of genes to include in the group given either as a single number or multiple numbers separated by

commas. The value(s) are used to pick those genes that are both significant overexpressed and underexpressed in each column of allGenesFile. If more than 3 values are specified the output will be an html file containing a plot of the p-values for the different values. Defaults to 100

--go org Include annotations from the Gene Ontology (http://www.geneontology.org/). Includes all the annotations for the genes achieved by traversing the Directed Acyclic Graph formed by all terms in the gene ontology. org represents the organism to use, can at the moment be human or yeast.

--graph Instead of making a text report create a graphical representation of the pvalues. If more than 3 values were specified for the -s option a html report will be created and a graph will be created for the calculation of max(s values)

EXAMPLES

1) Calculated the enrichment of annotations from the gene ontology and spellman cellcycle classified genes in 5 eigenarrays calculated using SVD. Creating a text report.

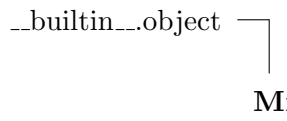
```
enrichcommand -s200 --go  eigenarrays_5.txt 1 yeast_cellcycle_spellman
```

```
2) Calculate the enrichments as above but generate a html report.

   enrichcommand -s50,100,150,200 --go  eigenarrays_5.txt 1
             yeast_cellcycle_spellman
```

A.2 Module MicroArray

A.2.1 Class MicroArray



Contains data from set of microarray experiments. Allows manipulations of data.

Example Usage:

```
>>> #Read in the data
>>> ma = MicroArray('path/to/tab/delimited/file', 2)

>>> #See size of data
>>> print ma.data.shape

>>> #Extract multiple experiments by specifying
>>> # the slide names or experiment names
>>> ma=ma.getExps(['yB12n099','yB12n100','yB12n138'])

>>> #if experimental names contain multiple conditions
>>> # we can split those labels
>>> ma.splitExpNames(r'(.*) vs. (.*) (P[0-9]*) (.*?) ([0-9]*\.?[0-9]*).*hr \
...    set *([0-9]*) *(.*)', 7)
```

```
>>> #Filter out all the genes or rows with missing values
>>> ma.filterNaN(1.0)

>>> #Normalize so that each array has the same range of expression values.
>>> ma.normArrayScale()

>>> #Save the output as a matlab readable and tab delimited file
>>> ma.saveMatlab('filename.mat')
>>> ma.saveTab('filaname.csv')
```

Methods

__init__(*self*, *p1=''*, *p2=''*, *p3=''*, *p4=''*)

Constructor called with two arguments: filename and nGeneAnots;

Parameters

`p1`: **filename** - points to a tab delimeted file in which the first row contains columm headers and the first nGeneAnots columns contain row headers.

`p2`: **nGeneAnots** - Number of columns of rowHeaders

Overrides: __builtin__.object.__init__

__getitem__(*self*, *k*)

__getslice__(*self*, *i*, *j*)

__setitem__(*self*, *i*, *value*)

__str__(*self*)

Overrides: __builtin__.object.__str__

averageRepeats(*self*, *requiredSame*)

Averages repeated experiments that are same according to classifications in requiredSame.

Parameters

`requiredSame:` sequence object of integers containing classifications that will have to be same. i.e. [1,3,4]

deleteExps(*self*, *idx*)

Removes experiments i.e. columns

Parameters

`idx:` Indexes of columns (experiments) to be removed.

filterNaN(*self*, *cutoff*)

Keeps Genes that have a percentage of non missing values greater than or equal to cuttoff

Parameters

`cutoff:` percentage of experiments that have to be good to keep.

filterNaNExp(*self*, *cutoff*)

Keeps Columns that have a percentage of non missing values less than or equal to cuttoff

Parameters

`cutoff:` percentage of experiments that have to be good to keep.

getExpIdx(*self*, *names*)

Searches through all expNames to find matches to NAMES. Where NAMES is either a

1. string, in which case the experiments which have names containing this string are found.
2. Sequence, in which case all experiments with names containing any of the strings in NAMES are found.

Return Value

a sequence if indexes

getExps(*self*, *names*)

Searches through all expNames to find matches to NAMES. Where NAMES is either a

1. string, in which case the experiments which have names containing this string are found.
2. Sequence, in which case all experiments with names containing any of the strings in NAMES are found.

Return Value

a new MicroArray object.

getGeneIdx(*self*, *names*)

Searches through all geneNames to find matches to NAMES. Where NAMES is either a

1. string, in which case the genes which have names containing this string are found.
2. Sequence, in which case all genes with names containing any of the strings in NAMES are found.

Return Value

a sequence if indexes

getGenes(*self*, *names*)

Searches through all geneNames to find matches to NAMES. Where NAMES is either a

1. string, in which case the experiments which have names containing this string are found.
2. Sequence, in which case all experiments with names containing any of the strings in NAMES are found.

Return Value

a new MicroArray object.

normArrayCenter(*self*)

Normalizes each array so that the average expression is 0. That is T_{:j} = T_{:j} - mean(T_{:,j}).

Example: ma.normArrayCener()

normArrayScale(*self*)

Normalizes each array so that the sum of expression squared is 1. That is T_{:j} = T_{:j}/sqrt(T_{:,j} \cdot T_{:,j}).

Example: ma.normArrayScale()

normFrobenius(*self*)

Normalizes the whole dataset such that the Frobenius norm is 1. That is $T=T/||T||_F$

Example: ma.normFrobenius()

replaceNaNInterp(*self*)

Replaces missing values in the matrix by linearly interpolating between existing values.

Example: The microarray:

```
[1, 2, 1, 3]
[2, 2, nan, 3]
[1, 4, 0, 2]
```

becomes:

```
[1, 2, 1, 3]
[2, 2, 2.5, 3]
[1, 4, 0, 2]
```

replaceNaNMean(*self*)

Replaces missing values in the matrix with the average expression value across all arrays for a specific gene.

Example: The microarray:

```
[1, 2, 1, 3]
[2, 2, nan, 3]
[1, 4, 0, 2]
```

becomes:

```
[1, 2, 1, 3]
[2, 2, 2.33, 3]
[1, 4, 0, 2]
```

replaceNaNSVD(*self*, *L*)

Replaces missing values in the matrix by estimates them as a least square supperposition of the L top eigenvectors of the row space.

Ref: Alter, O et. al. PNAS 100 (6), pp. 3351-3356 (March 2003)

Example: The microarray:

```
[1, 2, 1,   3]
[2, 2, nan, 3]
[1, 4, 0,   2]
```

when callled as ma.replaceNaNSVD(2) becomes:

```
[1, 2, 1,     3]
[2, 2, 1.087, 3]
[1, 4, 0,     2]
```

Parameters

`L:` Number of singular vectors to use in estimating missing values. Has to fullfill L<=min(nGenes, nExps).

saveMatlab(*self*, *filename*)

Saves microarray class as matlab readable file.

Parameters

`filename:` output file that will store tab delimited file.

saveTab(*self*, *filename*)

Saves microarray class as tab delimited file.

Parameters

`filename:` output file that will store tab delimited file

sortExp(*self*, *aOrdering*)

Sorts Microarrays by ordering experiments according to classifications given in sequence nOrdering. I.e. we sort first by classification in nOrdering[0] then without changing order of these we sort according to nOrdering[1] etc.

Parameters

`aOrdering:` sequence of classifications to sort after.

splitExpNames(*self*, *regexp*, *size*)

---

splitExpNames - Splits experimental name up into classifications

Each experimental name lists many criterea that was fullfilled for that experiments such as time point, cell type, chemical environment etc. In order to be able to sort experiments and filter simillar experiments this method takes a regular expression and populates MicroArray.expClass with the classifications in each column.

Example

Assume the experiment names look like:

```
[UHR vs. WI-38 P7 PDGF 4hr set 3
 UHR vs. WI-38 P8 PDGF 8hr set 1
 UHR vs. WI-38 P8 PDGF 8hr set 2
 UHR vs. WI-38 P7 PDGF 8hr set 3
 UHR vs. WI-38 P6 Serum 0 hr set 1
 UHR vs. WI-38 P6 Serum 0 hr set 1 hyb2]
```

Then the regular expression:

r'(.*) vs. (.*) (P[0-9]*) (.*?) ([0-9]*\.?[0-9]*).*hr set *([0-9]*) *(.*)'

will split these names into 7 classifications.

Parameters

`regexp`: the regular expression with () groupings around each type of classification

`size`: the number of classifciations

Inherited from object: __delattr__, __getattribute__, __hash__, __new__, __reduce__, __reduce_ex__, __repr__, __setattr__

A.3 Module EnrichProbability

A.3.1 Functions

binomial(*k*, *i*)

Returns the binomial coefficient of k choose i.

Parameters

`k:` Population size

`i:` Sample size

Return Value

k!/((k-i)!i!)

EnrichProbabilityFactory(*fileList*, *allGenes*)

Creates multiple EnrichProbability objects for multiple classifications.

Parameters

`fileList:` List of file pointers where each file is a tab delimited mapping from labels to classifications.

`allGenes:` List of all possible labels that can be chosen (i.e. list of all probes on array).

Return Value

List of `EnrichProbability` objects that can be queried for enrichment.

hypergeom(*N*, *K*, *m*, *l*)

Calculates the hypergeometric distribution.

Parameters

`N:` Size of population (e.g number of genes on array)

`K:` Number of successes within population (e.g. number of genes with specified attribute in whole genome)

`m:` Sample size (e.g. number of genes in cluster)

`l:` Number of successes within sample (e.g. number of genes with specified attribute in cluster)

Return Value

The hypergeometric cumulative distribution function from 1 to K.

$$\sum_{i=l+1}^{m} rac{{i N-K\choose}}{ N\choose}} $$

incDict(*dict*, *key*, *count*=`1`)

Increases the value at position key in the dictionary by 1. The dictionary has to be a mapping between keys and numbers.

Parameters

`dict:` Dictionary of key, value pairs where the values are integers.

`key:` Any valid key for the dictionary.

`count:` (Optional) number to increase value by. Defaults to 1.

A.3.2 Class EnrichProbability

Known Subclasses: GOEnrichProbability

Keeps track of annotations of labels and calculates enrichment numbers and p values using hypergeometric distribution of sub samples.

```
>>> ep = EnrichProbability.EnrichProbability(open('yeast_cellcycle_spellman'),
['YNR044W','YKL185W','YLR274W','YBR202W','YJL194W'])
>>> ep(['YKL185W', 'notinlist'])
[{'class': 'M/G1', 'nTot': 4, 'nr': [1], 'p': [1.0000000000000002]}]
>>> ep.getClassifications(['YLR274W'])
['YLR274W', 'M/G1']
```

Methods

__init__(*self*, *filep*, *allGenes*)

Constructor taking a file pointer and a list of all the genes.

Parameters

`filep:`	pointer to tab delimited file containing mappings between labels and annotations (e.g. genes and annotations)
`allGenes:`	List of labels(genes) used to filter out labels and annotations from filep.

__**call**__(*s*, *genes*, *nLimit*=`0`, *steps*=`[0]`)

Calculates the enrichment of a set of labels or genes using the annotations stored in the object.

Parameters

`genes:` A list of genes for which enrichment is to be computed.

`nLimit:` Optional parameter that specifies the minimum number of genes required of a certain type of annotation for its enrichment to be calculated. (Defaults to 0)

`steps:` Optional list of numbers used when wanting to examine subgroups of the genes. For each number n in steps the n top genes in the variable genes is examined and the enrichment calculated. The Default value is set so that all the genes are used.

Return Value

Returns a list of dictionaries where each dictionary has the following keys:

- **class:** the enrichment category.
- **nr:** a list of numbers indicating the number of genes with the class annotation. Each position corresponds to separate step.
- **nTot:** number of genes in genome or entire set that has annotation
- **p:** list of p-values for each of the steps.

getClassifications(*self*, *genes*)

Returns the classification of a series of genes.

Parameters

`genes:` List of genes for which the classifications should be returned.

Return Value

list of classifications for each gene in genes.

Class Variables

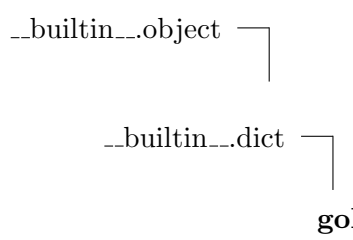
Name	Description
nGenes	**Value:** `0` *(type=`int`)*

A.4 Module goDAG

A.4.1 Functions

error(*s*)

A.4.2 Class goDAG



Keeps track of GO directed acyclic graph(DAG) by storing GO in dictionary of goTerms. Allows for different types of queries.

Methods

__init__(*self*, *filename*)

Creates a goDAG from OBO file.

Parameters

`filename:` name of OBO file as can be downloaded from *http://www.geneontology.org/ontology/gene_ontology_edit.obo*

Overrides: __builtin__.dict.__init__

extractAll(*self*, *id*)

Returns all parents of given id as a set.

Parameters

`id:` GO term id string in format 'GO:nnnnnnnn'

Return Value

set of all GO terms as GO id strings that are parents of id.

name2id(*self*, *name*)

Given a name as a string returns the equivalent GO id or None if name is not present in goDAG.

Parameters

`name:` String representing a GO term name.

printTree(*self*, *id*, *level*=0)

Prints to standard out goTerm tree given a term (id).

Parameters

`id:` GO term id of which all parent terms are to be output.

`level:` Parameter that determines number of indentations to use for parental levels. Should not be necessary to set by user. Defaults to 0.

Inherited from dict: __cmp__, __contains__, __delitem__, __eq__, __ge__, __getattribute__, __getitem__, __gt__, __hash__, __iter__, __le__, __len__, __lt__, __ne__, __new__, __repr__, __setitem__, clear, copy, get, has_key, items, iteritems, iterkeys, itervalues, keys, pop, popitem, setdefault, update, values

Inherited from object: __delattr__, __reduce__, __reduce_ex__, __setattr__, __str__

Inherited from type: fromkeys

A.4.3 Class GOEnrichProbability

Extension of EnrichProbability class that keeps track of GO classifications and traverses entire GO DAG upon queries.

Contains:

- **organism**: an organism_goDAG containing the information of the gene ontology and the mapping from gene names to go terms.

Methods

__**init**__(*self*, *filep*, *allGenes*)
Constructor taking a file pointer and a list of all the genes. **Parameters** `filep:` pointer to tab delimited file containing mappings between labels and annotations (e.g. genes and annotations) `allGenes:` List of labels(genes) used to filter out labels and annotations from filep. Overrides: EnrichProbability.EnrichProbability.__init__ extit(inherited documentation)

__**call**__(*self*, *genes*, *nLimit*=`0`, *steps*=`[0]`)
Overrides: EnrichProbability.EnrichProbability.__call__

Inherited from EnrichProbability: getClassifications

Class Variables

Name	Description
Inherited from EnrichProbability: nGenes *(p. 75)*	

A.4.4 Class goTerm

Class that keeps track of one gene ontology(GO) term including the following:

- **parents**: list of GO terms that are above this term in the Directed Acyclic Graph.
- **id**: the GO id (i.e. 'GO:nnnnnnnn')

- **name**: common name of term
- **definition**: longer definition describing term
- **namespace**: one of {BP,MF,CC} standing for Biological Process, Molecular Function or Cellular Compartment

Methods

__**init**__(*self*, *id*, *name*, *definition*, *namespace*, *parents*)
Constructor, sets instance variables.

__**str**__(*self*)
Returns string containing 'id name '

A.4.5 Class organism_goDAG

Has an organism mapping and a goDAG allowing for queries of classification of genes.

Contains two instance variables:

- **sgdDict**: is a mapping from gene names and a set of GO terms
- **goTerms**: is the directed acyclic graph of GO terms

Methods

__**init**__(*self*, *filename*)

Reads the conversion between gene names and GO terms

Parameters

`filename:` file that contains mappings from genes to GO terms. The file for mapping yeast genes can be found at *http://www.geneontology.org/GO.current.annotations.shtml*

genes2AnnotDict(*self*, *genes*)

Given a list of genes returns all the annotations and the number of genes of each kind of annotation.

Parameters

`genes:` List of gene names.

Return Value

dictionary of key, value pairs where the keys are the annotations and the values are the number of times that annotation appears in the genes list.

id2genes(*self*, *id*)

Returns all the genes that are annotated as the specified id.

Parameters

`id:` GO id ('GO:nnnnnnnn')

Return Value

List of gene names.

name2genes(*self*, *name*)

Returns all the genes that are annotated by the GO term with given name.

Parameters

`name:` GO term name.

Return Value

List of gene names.

A.5 Module sparklines

A.5.1 Functions

plot_sparkline(*x*, *y*, *args*)

Returns a sparkline image as a data: URI.

`>>> sparklines.plot_sparkline([1,2,3,4,5], [.1, .2, .1, .2,.5], {})`

Many thanks to Joe Gregorio *http://bitworking.org/projects/sparklines/* for inspiration.

Parameters

`x:` list of positions on x-axis

`y:` list of positions on y-axis

`args:` dictionary where the following keys can be set (default values in parenthesis):

- **height:** number of pixels of image (height=20)
- **limits:** min and max value of y to be plotted (limits=[min(y), max(y))
- **scale:** log or linear scaling of y axis (scale=linear)
- **step:** number of pixels each datapoint takes up (step=1)
- **hasMin:** show the maximum coordinate (true)
- **hasMax:** show the minimum coordinate (true)
- **hasLast:** show the last coordinate (true)

Return Value

A URI representing a png image that can be incorporated as a string within a html document.

Appendix B

Data Sources for Gene Annotations

The gene annotations were culled from literature. Here is presented in detail how these annotations were extracted.

B.1 Yeast Data

B.1.1 Cellcycle Stages Classification

In Spellman's paper [56] 800 genes were classified into the cell-cycle stages M/G1, G1, S, S/G2 and G2/M using clustering of microarray data. While traditional methods have classified 103 genes into the same categories. The genes from Spellman were extracted from figure 1 from the above paper. Since the division of the 800 cellcycle genes into the five different groups termed G1, S, G2/M, S/G2, and M/G1 was somewhat arbitrary according to Spellman the groups were also divided into I_3 consisting of those classified as G2/M and M/G1 by Spellman; S consisting of S/G2

and S; G2 consisting of those classified as G2/M and S/G2; G1 consisting of M/G1 and G1 genes.

The paper[56] also classifies small groups of genes into cellcycle controlled clusters (ALPHA, CLB2, CLN2, Histones, MAT, MCM, MET, SIC1 and Y'). These clusters were extracted from the supllementary figures available on the website http://genome-www.stanford.edu/cellcycle/figures/. Zhu et al. [69] also used these classifications but with slightly different genes in each cluster. The clusters for MCM, SIC1, CLB2, CLN2 and histones were extracted from figure 3 of the paper.

B.1.2 Stress Response

Gasch et al. [26] studied the stress response of yeast under many conditions and identified the environmental stress response (ESR) that consists of 285 genes that are up regulated by stress and 583 genes that are down regulated. These genes were downloaded as the data for figure 3 from the supplementary website at http://www-genome.stanford.edu/yeast_stress

B.1.3 Pheromone Response

The genes that are up regulated and down regulated by pheromones were identified in [51] using microarray studies.

B.1.4 Origin of Replication Location Analysis

Wyrick et al. [67] carried out a CHIP-Chip study for the binding sites for the Origin Recognition Complex (ORC) and minichromosome maintenance (MCM) proteins in order to identify hypothetical DNA replication origins. The genes associated with these binding sites was determined by extracting the binding sites from the raw

data published on the supplemental website http://web.wi.mit.edu/young/origins/. This was done by assuming that a gene is bound to the gene if the p-value of binding is ≤ 0.005. Additional genes were assigned binding if intergenic regions upstream of the gene had significant bind as per the supplementary information at http://jura.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=17&f=geneassign [39].

In addition MacAlpine and Bell [40] have published a list of genes that they call Array Based Origins (ABO) which are origins that show consensus binding across three different microarrays studies. This list was downloaded from http://bell-lab-server.mit.edu/ABOrimaps/ and was supplemented by doing the same intergenic to gene mapping as above.

B.1.5 Transcription Factor Binding location

We also included the binding data of transcription factors as determined using CHIP-chip analysis carried out at the same lab as the origin binding [28]. The raw data was downloaded from the supplemental website http://web.wi.mit.edu/young/regulatory_code and each gene that had a p-value < 0.001, as in the paper, was determined bound by the transcription factor.

Bibliography

[1] O Alter, P O Brown, and D Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A*, 97(18):10101–10106, Aug 2000. 4, 17, 27, 54

[2] O. Alter, P.O. Brown, and D. Botstein. Processing and modeling genome-wide expression data using singular value decomposition. *Proc. SPIE*, 4266(2):171–186, 2001. 17

[3] O. Alter and G.H. Golub. Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations. *Proceedings of the National Academy of Sciences*, 102(49):17559–17564, 2005. 3

[4] O. Alter and G.H. Golub. Singular value decomposition of genome-scale mRNA lengths distribution reveals asymmetry in RNA gel electrophoresis band broadening. *Proceedings of the National Academy of Sciences*, 103(32):11828, 2006. 4

[5] Orly Alter. Discovery of principles of nature from mathematical modeling of DNA microarray data. *Proc Natl Acad Sci U S A*, 103(44):16063–16064, October 2006. 1, 2

[6] Orly Alter, Patrick O Brown, and David Botstein. Generalized singular value

decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proc Natl Acad Sci U S A*, 100(6):3351–3356, Mar 2003. 2

[7] Orly Alter and Gene H Golub. Integrative analysis of genome-scale data by using pseudoinverse projection predicts novel correlation between DNA replication and RNA transcription. *Proc Natl Acad Sci U S A*, 101(47):16577–16582, Nov 2004. 2, 5

[8] F. Blanchard, M.E. Rusiniak, K. Sharma, X. Sun, I. Todorov, M.M. Castellano, C. Gutierrez, H. Baumann, and W.C. Burhans. Targeted Destruction of DNA Replication Protein Cdc6 by Cell Death Pathways in Mammals and Yeast. *Molecular Biology of the Cell*, 13:1536–1549, 2002. 5, 54

[9] R. Bro. Parafac: tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38:149–171, 1997. 4, 34

[10] R. Bro and H.A.L. Kiers. A new efficient method for determining the number of components in PARAFAC models. *Contract*, 1999:10377. 37, 55

[11] Rasmus Bro. *Multi-way Analysis in the Food Industry: Models, Algorithms and Applications*. PhD thesis, Royal Veterinary and Agricultural University, 1998. 3, 4, 36, 37

[12] D.T. Burhans, L. Ramachandran, J. Wang, P. Liang, H.G. Patterton, M. Breitenbach, and W.C. Burhans. Non-random clustering of stress-related genes during evolution of the S. cerevisiae genome. *BMC Evolutionary Biology*, 6(1):58, 2006. 5, 55

[13] J.D. Carroll and J.J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970. 34

[14] JM Cherry, C. Ball, S. Weng, G. Juvik, R. Schmidt, C. Adler, B. Dunn, S. Dwight, L. Riles, RK Mortimer, et al. Genetic and physical maps of Saccharomyces cerevisiae. *Nature*, 387(6632 Suppl):67–73, 1997. 5, 55

[15] J.H. Cocker, S. Piatti, C. Santocanale, K. Nasmyth, and J.F.X. Diffley. An essential role for the Cdc 6 protein in forming the pre-replicative complexes of budding yeast. *Nature*, 379(6561):180–182, 1996. 5, 54

[16] Gene Ontology Consortium. The Gene Ontology (GO) project in 2006. *Nucleic Acids Research*, 34:D322–D326, 2005. 5, 53

[17] FH CRICK. On protein synthesis. *Symp Soc Exp Biol*, 12:138–63, 1958. 8

[18] Lieven de Lathauwer, Bart de Moor, and Joos Vandewalle. A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Anal. Appl.*, 21(4):1253–1278, 2000. xii, 4, 30, 31

[19] J. Demeter, C. Beauheim, J. Gollub, T. Hernandez-Boussard, H. Jin, D. Maier, J.C. Matese, M. Nitzberg, F. Wymore, Z.K. Zachariah, et al. The Stanford Microarray Database: implementation of new analysis tools and open source release of software. *Nucleic Acids Research*, 35(Database issue):D766, 2007. 13

[20] John F.X. Diffley, Julie H. Cocker, Simon J. Dowell, and Adele Rowley. Two steps in the assembly of complexes at yeast replication origins in vivo. *Cell*, 78(2):303–316, jul 1994. 5, 54

[21] J.J. Donato, S.C.C. Chung, and B.K. Tye. Genome-Wide Hierarchy of Replication Origin Usage in Saccharomyces cerevisiae. *PLoS Genet*, 2(9):e141, 2006. 5, 54

[22] J.A. Downs and S.P. Jackson. Involvement of DNA End-Binding Protein Ku in Ty Element Retrotransposition. *Molecular and Cellular Biology*, 19(9):6260–6268, 1999. 5, 53

[23] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 26

[24] G. Feten, T. Almøy, and A.H. Aastveit. Prediction of Missing Values in Microarray and Use of Mixed Models to Evaluate the Predictors. *Statistical Applications in Genetics and Molecular Biology*, 4(1):10, 2005. 15, 16

[25] J A Flattery-O'Brien and I W Dawes. Hydrogen peroxide causes RAD9-dependent cell cycle arrest in G2 in Saccharomyces cerevisiae whereas menadione causes G1 arrest independent of RAD9 function. *J Biol Chem*, 273(15):8564–8571, Apr 1998. 5, 52

[26] A P Gasch, P T Spellman, C M Kao, O Carmel-Harel, M B Eisen, G Storz, D Botstein, and P O Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular biology of the cell*, 11(12), 2000. 2, 41, 87

[27] G.H. Golub and C.F. Van Loan. *Matrix computations*. Johns Hopkins University Press Baltimore, MD, USA, 1996. 4, 32

[28] Christopher T Harbison, D Benjamin Gordon, Tong Ihn Lee, Nicola J Rinaldi, Kenzie D Macisaac, Timothy W Danford, Nancy M Hannett, Jean-Bosco Tagne, David B Reynolds, Jane Yoo, Ezra G Jennings, Julia Zeitlinger, Dmitry K Pokholok, Manolis Kellis, P Alex Rolfe, Ken T Takusagawa, Eric S Lander, David K Gifford, Ernest Fraenkel, and Richard A Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, Sep 2004. 2, 11, 41, 88

[29] RA Harshman. Foundations of the PARAFAC procedure: Models and methods for an "explanatory" multi-mode factor analysis. Technical report, UCLA Working Papers in Phonetics, 16, 1-84, 1970. 34

[30] J. Hastad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644654, 1990. 28

[31] VR Iyer, CE Horak, CS Scafe, D. Botstein, M. Snyder, and PO Brown. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, 409(6819):533–8, 2001. 11

[32] H.A.L. Kiers. Hierarchical relations among three-way methods. *Psychometrika*, 56(3):449–470, 1991. 34

[33] P.J. Killion, G. Sherlock, and V.R. Iyer. The Longhorn Array Database (LAD): An Open-Source, MIAME compliant implementation of the Stanford Microarray Database (SMD). *feedback*, 2004. 13

[34] H. Kim, G.H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187–198, 2005. 15

[35] TG Kolda. Orthogonal Tensor Decompositions SIAM Journal on Matrix Analysis and Applications. *Vol*, 23:243–255, 2001. 4

[36] M. Larochelle, S. Drouin, F. Robert, and B. Turcotte. Oxidative Stress-Activated Zinc Cluster Protein Stb5 Has Dual Activator/Repressor Functions Required for Pentose Phosphate Pathway Regulation and NADPH Production. *Molecular and Cellular Biology*, 26(17):6690–6701, 2006. 5, 54

[37] M. Le Romancer. Cleavage and inactivation of DNA-dependent protein kinase catalytic subunit during apoptosis in Xenopus egg extracts, 1996. 5, 53

[38] SE Lee, JK Moore, A Holmes, Umezu K, RD Kolodner, and JE Haber. "saccharomyces ku70, mre11/rad50, and rpa proteins regulate adaptation to g2/m arrest after dna damage". *Cell*, 94(3):399–409, Aug 1998. 5, 53

[39] T.I. Lee, N.J. Rinaldi, F. Robert, D.T. Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannett, C.T. Harbison, C.M. Thompson, I. Simon, et al. Transcriptional Regulatory Networks in Saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002. 88

[40] D.M. MacAlpine and S.P. Bell. A genomic view of eukaryotic DNA replication. *Chromosome Research*, 13(3):309–326, 2005. 88

[41] E.R. Mardis. ChIP-seq: welcome to the new frontier. *Nature Methods*, 4:613–614, 2007. 2

[42] G. Micklem, A. Rowley, J. Harwood, K. Nasmyth, and J.F.X. Diffley. Yeast origin recognition complex is involved in DNA replication and transcriptional silencing. *Nature*, 366(6450):87–89, 1993. 5, 54

[43] DT Nguyen, AM Alarco, and M. Raymond. Multiple Yap1p-binding sites mediate induction of the yeast major facilitator FLR1 gene in response to drugs, oxidants, and alkylating agents.(2001). *J. Bio. Chem*, 276(2):1138–1145, 2001. 5, 53

[44] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003. 16

[45] L. Omberg, G.H. Golub, and O. Alter. A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. *Proceedings of the National Academy of Sciences*, 104(47):18371–18376, 2007. vi, 39

[46] E. Petricoin, J. Wulfkuhle, V. Espina, and L.A. Liotta. Clinical proteomics: revolutionizing disease detection and patient tailoring therapy. *J Proteome Res*, 3(2):209–17, 2004. 2

[47] J.R. Pollack, C.M. Perou, A.A. Alizadeh, M.B. Eisen, A. Pergamenschikov, C.F. Williams, S.S. Jeffrey, D. Botstein, and P.O. Brown. Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nature Genetics*, 23:41–46, 1999. 11

[48] J.R. Pollack, T. Sorlie, C.M. Perou, C.A. Rees, S.S. Jeffrey, P.E. Lonning, R. Tibshirani, D. Botstein, A.L. Borresen-Dale, and P.O. Brown. Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. *Proceedings of the National Academy of Sciences*, 99(20):12963–12968, 2002. 11

[49] J. Quackenbush. Microarray data normalization and transformation. *Nature Genetics*, 32(supp):496–501, 2002. 18

[50] L. Ramachandran, D.T. Burhans, P. Laun, J. Wang, P. Liang, M. Weinberger, S. Wissing, S. Jarolim, B. Suter, F. Madeo, et al. Evidence for ORC-dependent repression of budding yeast genes induced by starvation and other stresses. *FEMS Yeast Research*, 6(5):763–776, 2006. 5, 55

[51] C J Roberts, B Nelson, M J Marton, R Stoughton, M R Meyer, H A Bennett, Y D He, H Dai, W L Walker, T R Hughes, M Tyers, C Boone, and S H Friend. Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles. *Science*, 287(5454):873–880, Feb 2000. 2, 41, 87

[52] D.T. Scholes, M. Banerjee, B. Bowen, and M.J. Curcio. Multiple Regulators of Ty1 Transposition in Saccharomyces cerevisiae Have Conserved Roles in Genome Maintenance. *Genetics*, 159(4):1449–1465, 2001. 5, 53

[53] Michael Shapira, Eran Segal, and David Botstein. Disruption of yeast forkhead-associated cell cycle transcription by oxidative stress. *Mol Biol Cell*, 15(12):5659–5669, Dec 2004. 2, 4, 5, 40, 52

[54] I. Simon, J. Barnett, N. Hannett, C.T. Harbison, N.J. Rinaldi, T.L. Volkert, J.J. Wyrick, J. Zeitlinger, D.K. Gifford, T.S. Jaakkola, et al. Serial Regulation of Transcriptional Regulators in the Yeast Cell Cycle. *Cell*, 106(6):697–708, 2001. 2, 41

[55] M. Snyder, RJ Sapolsky, and RW Davis. Transcription interferes with elements important for chromosome maintenance in Saccharomyces cerevisiae. *Molecular and Cellular Biology*, 8(5):2184–2194, 1988. 5, 54

[56] P T Spellman, G Sherlock, M Q Zhang, V R Iyer, K Anders, M B Eisen, P O Brown, D Botstein, and B Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12):3273–3297, Dec 1998. xi, 2, 4, 16, 40, 86, 87

[57] A.C. Syvänen. Toward genome-wide SNP genotyping. *Nat Genet*, 37:5–10, 2005. 11

[58] S Tavazoie, J D Hughes, M J Campbell, R J Cho, and G M Church. Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–285, Jul 1999. 4

[59] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999. 21

[60] M.T. Teh, D. Blaydon, T. Chaplin, N.J. Foot, S. Skoulakis, M. Raghavan, C.A. Harwood, C.M. Proby, M.P. Philpott, B.D. Young, et al. Genomewide Single Nucleotide Polymorphism Microarray Mapping in Basal Cell Carcinomas Unveils Uniparental Disomy as a Key Somatic Event, 842. 11

[61] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R.B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001. 15, 16

[62] L.R. Tucker. The extension of factor analysis to three-dimensional matrices. *Contributions to Mathematical Psychology*, pages 109–127, 1964. 31

[63] L.R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966. 31

[64] E.R. Tufte. *Beautiful evidence*. Graphics Press, 2006. 23

[65] Wikipedia. Genomics — wikipedia, the free encyclopedia, 2007. [Online; accessed 17-September-2007]. 9

[66] Silke Wissing, Paula Ludovico, Eva Herker, Sabrina Bttner, Silvia M. Engelhardt, Thorsten Decker, Alexander Link, Astrid Proksch, Fernando Rodrigues, Manuela Corte-Real, Kai-Uwe Frhlich, Joachim Manns, Cline Cand, Stephan J. Sigrist, Guido Kroemer, , and Frank Madeo. An AIF orthologue regulates apoptosis in yeast. *J Cell Bio*, 166(7):969–974, Sep 2006. 5, 53

[67] J.J. Wyrick, J.G. Aparicio, T. Chen, J.D. Barnett, E.G. Jennings, R.A. Young, S.P. Bell, and O.M. Aparicio. Genome-Wide Distribution of ORC and MCM Proteins in S. cerevisiae: High-Resolution Mapping of Replication Origins. *Science*, 294(5550):2357, 2001. 2, 41, 87

[68] T. Zhang and GH Golub. Rank-1 approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl*, 23:534–550, 2001. 4

[69] G Zhu, P T Spellman, T Volpe, P O Brown, D Botstein, T N Davis, and B Futcher. Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature*, 406(6791):90–94, Jul 2000. 87

Vita

Larsson Omberg, son of Camilla and Gunnar Omberg, was born on February 11th, 1977 in Ljusne, Sweden. At the age of 9 he immigrated with his family to Portugal where he attended the American International School before graduating while an exchange student at Pinewood Preparatory School in Charleston, SC. In 1996 he enrolled at the Royal Institute of Technology in Sweden where he, in 1999, graduated with a M.Sc in Engineering Physics. After spending almost two years teaching he began his graduate work at the University of Texas at Austin in January 2001.

Permanent Address: 1605 Walnut Ave.
Austin, TX 78705

This dissertation was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.